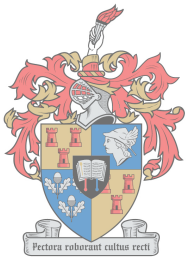
Implementation of the Fire Beam Element (FBE) in OpenSees for the analysis of structures in fire

by

Janeke Franziska Volkmann



UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

Thesis presented in partial fulfilment of the requirements for the degree of Masters in Engineering in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr Richard S. Walls

December 2018

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ..... December 2018 .....

Abstract

Fire safety design for buildings has become an increasing concern worldwide, especially after disasters such as the World Trade Centre collapse and the more recent Grenfell Tower fire. In such cases, structural fire engineering plays a crucial role in the inherent fire resistance of a structure. Traditionally, prescriptive methods were used to achieve fire resistance. These methods, however, are largely based on single-element tests in a standard fire and therefore do not consider critical global building behaviour. Consequently, performance-based design methods are becoming accepted as the more scientific approach to the design of structures in fire.

The Fire Beam Element (FBE) method is such a performance-based approach that simplifies a structure into a skeletal frame consisting of only beam and column elements. The work in this thesis employed the FBE method, modelled it in the OpenSees software environment, and subsequently applied it the context of global structures with restraint.

To validate the developed model, benchmark case studies were sourced from literature and conducted in order to ensure that the model behaved accordingly. These case studies comprised a steel beam with high axial and bending forces, two simply-supported composite beams and a 2D steel frame, respectively. The second case study confirmed that even composite cross-sections can be modelled using a single beam element — the model produced results similar to that of more a complex 3D shell model. The final case study demonstrated that complex behaviour such as non-linear heating, restraint and buckling were captured by the model — the latter two phenomena have not been considered in previous FBE work.

A sensitivity analysis of the FBE model found that the effect of increasing the mesh size of a cross-section does not increase the computational run time exponentially, thereby illustrating the developed models ability to handle more complex analyses. Furthermore, the efficiency of the FBE model allowed for a temperature sensitivity study to be conducted, and as temperature changes have significant effect on structural behaviour, this would potentially be useful in design. In future, if the FBE method is developed in 3D, it could be linked to the Slab Panel Method, and presented as a possible analysis tool for structural fire design of entire buildings.

Opsomming

Brandveiligheidsontwerp in geboue is wêreldwyd 'n toenemende bekommernis, veral na rampe soos die Wêreldhandelsentrum-ineenstorting en die meer onlangse Grenfell-toring brand. In sulke gevalle speel strukturele brandontwerp 'n deurslaggewende rol in die inherente brandweerstand van 'n struktuur. Tans gebruik struktuuringenieurs hoofsaaklik voorskriftelike metodes in die ontwerp van brandbestandheid. Hierdie metodes word egter hoofsaaklik gebaseer op enkel-element toetse in 'n standaard vuur en beskou dus nie kritiese globale bougedrag nie. Gevolglik word prestasiegebaseerde ontwerpmetodes aanvaar as die meer wetenskaplike benadering tot die ontwerp van brandende strukture.

'n Voorbeeld van so 'n prestasie-gebaseerde ontwerp benadering is die Brand-Balk Element (BBE) metode. Die BBE vereenvoudig 'n struktuur deur dit in 'n skeletale raam te verander wat net uit balke en kolomme bestaan. Die werk in hierdie tesis het die BBE metode gebruik, gemodelleer in die OpenSees sagteware-omgewing en daarna is dit in die konteks van ingeklemde strukture toegepas.

Om die ontwikkelde model te valideer, is standaard experimente van literatuur verkry en uitgevoer om te verseker dat die model homself dienooreenkomstig gedra. Drie sulke gevallestudies was uitgevoer: 'n staal balk met 'n hoë aksiale krag en buigkragte, twee saamgestelde eenvoudig-opgelegde balke en 'n 2D staal raamwerk. Die tweede eksperiment het bevestig dat selfs saamgestelde dwarssnitte met 'n enkele balk element gemodelleer kan word — die model het resultate soortgelyk aan dié van meer komplekse 3D-dopmodelle gelewer. Die finale maatstafeksperiment het getoon dat komplekse gedrag soos nie-lineêre verwarming, ingeklemdheid en buiging deur die model vasgevang word. Die laasgenoemde twee verskynsels is nie in vorige BBE werk oorweeg nie.

'n Sensitiwiteitsanalise van die BBE model het bevind dat die effek van die vergroting van die maaswydte van 'n deursnit nie die model se berekeningstydperk eksponensieel verhoog het nie, en sodoende is die ontwikkelde modelle se vermoë om meer komplekse ontledings te hanteer geïllustreer. Die eenvoudigheid van die BBE model laat toe om verskillende temeratuur profiele te toets en is dus gerieflik vir ontwerp doeleindes. In die toekoms, as die BBE metode in 3D ontwikkel word, kan dit in samewerking met die Paneelmetode gebruik work, en aangebied word as 'n moontlike analise-instrument vir strukturele brandontwerp van hele geboue.

Acknowledgements

I would like to thank the following people:

- Dr Richard Walls, for your never-ending enthusiasm, patience, willingness to help at all times, quick email-responses and thorough feedback.
- The Wilhelm Frank Trust, for the funding and the opportunity.
- My family: My parents, for your love and support all the way from Namibia; my brother Sven, for always finding a way to make my day (especially when his girlfriend Lillian sends treats just before hand-in time); my aunt Edda and my cousins, for being my support structure here in Stellenbosch.
- My best friends Mattie, Esmé and Petro. Thank you for listening to daily updates on my thesis during all the lunch breaks and coffee sessions.
- Anro, for your time and effort in winning the war against the bugs for me. Ryan, for your lessons in language and writing style.
- And finally, to my fellow officers: Gloomy, for downloading and installing all of the software; Grumpy, for your proofreading, translating and old-man's advice; and Pookie for your company and rusks during late-night working sessions.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

2D Two-Dimensional

3D Three-Dimensional

ASCE American Society of Civil Engineers

BSI British Standards Institution

CCTV China Central Television

CEFICOSS Computer Engineering of the Fire design in Composite and Steel Structures

COST European Cooperation in the field of Scientific and Technical Research

CSV Comma-Separated Values

DOF Degree of Freedom

EN European Norm

FBE Fire Beam Element

FE Finite Element

FEA Finite Element Analysis

FEM Finite Element Method

FRR Fire Resistance Rating

ISO International Organization for Standardization

NA Neutral Axis

NIST National Institute for Science and Technology

RA Reference Axis

PEER Pacific Earthquake Engineering Research

SPM Slab Panel Method

TCL Tool Command Language

UDL Uniformly Distributed Load

WTC World Trade Center

XML Extensible Markup Language

Y.S. Yield stress

Symbols

Roman

a Parameter 1 for calculation of EN 1993-1-2 stress-strain curve of steelwork

A Cross-sectional area

b Parameter 2 for calculation of EN 1993-1-2 stress-strain curve of steelwork

c Parameter 3 for calculation of EN 1993-1-2 stress-strain curve of steelwork

c' Distance of neutral axis from reference axis

e Eccentricity of the force to the middle of the compression block

E Young's Modulus

E_{20} Young's Modulus at ambient temperature

E_0 Initial Young's Modulus

E_S Secant Young's Modulus

E_T Tangent Young's Modulus

E_θ Young's Modulus at temperature θ

EA Axial stiffness

EI Bending stiffness

$f_{c,\theta}$ Compressive strength of concrete at temperature θ

f_{ck} Characteristic cylinder strength of concrete in compression

$f_{p,\theta}$ Stress at the proportional limit at temperature θ

f_u Ultimate strength of steelwork at ambient temperature

f_y Yield stress of steelwork at ambient temperature

$f_{y,\theta}$ Yield stress at temperature θ

$\{\boldsymbol{f}\}$ Local load vector

F Force

$\{\boldsymbol{F}\}$ Global load vector

$\{\boldsymbol{F_r}\}$ Resisting force vector

$\{\boldsymbol{F_U}\}$ Unbalanced force vector

$\{\boldsymbol{F_\theta}\}$ Thermal load vector

$\{\boldsymbol{F_\sigma}\}$ Mechanical load vector

i Element or fibre number

I Second moment of inertia

j Iteration number

k Spring stiffness

$k_{c,t}$ Reduction factor for concrete strength in tension

$k_{E,\theta}$ Slope of linear elastic range reduction factor

$k_{p,\theta}$ Proportional limit reduction factor

$k_{y,\theta}$ Effective yield strength reduction factor

$[\boldsymbol{k}]$ Local stiffness matrix

$[\boldsymbol{k_{el}}]$ Local elastic stiffness matrix

$[\boldsymbol{k_g}]$ Local geometric stiffness matrix

$[\boldsymbol{K}]$ Global stiffness matrix

L Length of beam

n Total number of elements or fibres

N Axial load

N_θ Equivalent thermal axial force

M Applied Moment

M_θ Equivalent thermal moment

P Axial force

t Time in minutes

$[\boldsymbol{Q}]$ Matrix operator applied to the stiffness matrix to account for the neutral axis eccentricity

T Temperature

T Thrust reaction force

$[\boldsymbol{T_i}]$ Transformation matrix from local coordinate system of element i to global coordinate system

u displacement in the x direction

$\{\boldsymbol{u}\}$ Local displacement vector

v Displacement in the y direction

$[\boldsymbol{W}]$ Matrix operator applied to the load vector to account for the neutral axis eccentricity

$\{\boldsymbol{x}\}$ Position vector

y Distance from the NA to the centroid of a fibre element

Greek

α Coefficient of thermal expansion

α Nodal rotation

δ Deflection

Δ* Change of property *

ΔT Temperature difference from ambient temperature

$\{\boldsymbol{\Delta}\}$ Global displacement vector

$\{\boldsymbol{\Delta u}\}$ Displacement increment vector

ε Total strain

ε_{axial} Strain caused by an axial load

$\varepsilon_{bending}$ Strain caused by bending

$\varepsilon_{c1,\theta}$ Strain of concrete at maximum stress

ε_{creep} Creep strain

$\varepsilon_{cu1,\theta}$ Strain of concrete at failure

$\varepsilon_{t,\theta}$ Limiting strain for yield strength at temperature θ

ε_{tr} Transient strain

$\varepsilon_{p,\theta}$ strain at the limit of proportionality at temperature θ

$\varepsilon_{u,\theta}$ ultimate strain at temperature θ

$\varepsilon_{y,\theta}$ yield strain at temperature θ

ε_σ Mechanical strain

ε_θ Thermal strain

θ Temperature

θ_a Temperature of steelwork

θ_c Temperature of concrete

θ_g Gas temperature

σ Stress

σ_θ Equivalent thermal stresses

φ Rotation

Sub- and Superscripts

1 First node on the reference axis

2 Second node on the reference axis

20 Ambient temperature

a Steelwork

A First node on the neutral axis

B Second node on the neutral axis

b Matrix in the basic format, typically with only 3 rows

c Concrete

E Young's Modulus

el Elastic

g Geometric

i Element or fibre number

j Iteration number

p Proportional limit for steelwork

n Total number of elements or fibres

r Resisting

s Secant

U Unbalanced

x Major axis

y Yield strength

y Minor axis

θ At elevated temperature

θ Thermal

σ Mechanical

z Perpendiculat axis to the minor and major axes

Chapter 1

Introduction

1.1 Background to study

Structural fire engineering safety in buildings has become an increasing concern worldwide, as building become progressively larger, more complicated and new construction materials are used. Traditionally, the fire design of a structure is based on isolated single element testing under a standard fire. Passive protection is specified to ensure that steel members do not heat up to more than specified limits, concrete members have sufficient rebar cover or timber members have charring limited to a specific depth. However, global structural behaviour, member interactions, restraint, boundary conditions and other such factors are ignored in this process. Researchers in the field of structural fire engineering agree that there is a need for the development of new design methods that reflect structural response to fire in reality (Wang *et al.*, 2012; Jiang and Usmani, 2013). As a result, performance-based design approaches have been developed since the early 1970s. This rational design approach to fire engineering is based on scientific and engineering principles and aims to develop safer solutions (Wang *et al.*, 2012), while promoting innovation and cost-saving (Duthinh, 2014).

1.2 Problem statement

The existing performance-based design methods use advanced calculation models such as those developed in SAFIR (Franssen, 2005) and Vulcan (Burgess, 2015). The drawback of these complex models are that they demand a high level of technical skill from the designer (Wang *et al.*, 2012), as well as computational effort and user-input time. Such approaches are therefore not feasible for all structures and are mainly reserved for high-profile buildings, such as the China Central Television (CCTV) building in Beijing (Wang *et al.*, 2012). There is therefore a niche to develop performance-based design approaches that would be simpler and more feasible to apply to general structures. The slab panel method (SPM) developed by Clifton (2006) is an example of an intermediate level application to fire analyses for slabs. The work of Walls (2016) addressed this need for frames by developing the Fire Beam Element (FBE) methodology for a simpler design approach to structures in fire. However, the FBE method requires further research in its application when

considering global analysis and structural continuity. In the previous work it was only applied to simply-supported members, and iterative, non-linear analyses considering global behaviour could not be carried out. The research in this thesis therefore expands on the FBE concept by developing the implementation into finite element (FE) software. This has the overall aim to make performance-based fire design an accessible, user-friendly and feasible analysis tool for the design of all structures, contributing to general fire safety in buildings.

1.3 Research objectives

The main objective of this thesis is to develop the Fire Beam Element (FBE) methodology as a performance-based design approach to the global analysis of structures subjected to fire. The aims are as follows:

- apply the basic theory of FBE from first principles to structures with restraint
- integrate the FBE formulation into FE software, with OpenSees (Jiang and Usmani, 2013) being utilised
- developing user-friendly commands for the FBE method in FE software with the aim of providing an analysis tool for structural fire analysis in industry
- validate the FBE model by comparing it to results from numerical and experimental studies
- perform parametric studies which investigates the effect of temperature and mesh size on the FBE analysis model developed as these parameters typically play an important role in FE fire analyses

1.4 Scope

This research adopts the decoupled approach and focuses on the structural response of a fire analysis. The initial step of a decoupled fire analysis is a heat transfer analysis, but this is a complex research area in itself and is out of the scope of this research. Rather, the temperatures of the structural elements are assumed to be known and provided as input. The difference between a coupled and decoupled analysis is further explained in Chapter 2.

The implementation of the FBE methodology in this work only considers single axis bending and global 2D structures. Future work is required to extend the formulation to 3D models. However, the effect of torsion and minor-axis bending in the FBE model would have to be investigated.

The formulation of the beam element is based on Bernoulli's assumption that plane sections remain plane, as in most beam element formulations. Therefore, the FBE model becomes less accurate when plane-strain assumptions (i.e. no shear distortions) are violated, for example in the analysis of beams with multiple web openings.

This research has adopted the material models from the Eurocodes EN 1993-1-2 (BSI, 2005) for steel and EN 1992-1-2 (BSI, 2004*b*) for concrete. Tension stiffening in reinforced concrete has been investigated in the original work of FBE but found to only have limited influence on deflection predictions, and has therefore not been addressed in this work.

1.5 Thesis outline

Chapter 1 - Introduction
The first chapter provides background to performance-based design and presents the problem statement. Thereafter the objectives, scope and limitations of this work are outlined.

Chapter 2 - Literature review
The background knowledge required for this research is outlined. This includes the topics fire safety, structural fire engineering, fire models, material models of steel and concrete in fire, structural fire analysis and software used for fire design.

Chapter 3 - Theory of the Fire Beam Element methodology
The theory of the FBE methodology from first principles is provided. Importantly, the concept of a shifting neutral axis (NA) position relative to a reference axis (RA) is introduced and illustrated by a simple example. This is followed by an explanation of how the FBE concept can be integrated into FE theory. Lastly, the benefits and limitations of the FBE method are identified.

Chapter 4 - The implementation of the Fire Beam Element methodology into the OpenSees finite element software
This chapter explains how the FBE methodology was implemented in the FE software OpenSees for the modelling of global structures in fire. Firstly, OpenSees is introduced and the framework outlined. This is followed by a detailed explanation of how the FBE concept was developed in the OpenSees source code. Furthermore, the application of the FBE formulation for an OpenSees user is provided.

Chapter 5 - Validation of the FBE model by numerical and experimental studies
This chapter discusses three case studies which were investigated for the validation of the FBE model in OpenSees. The case studies provide numerical and experimental predictions of structural behaviour which are used as a comparison to the predictions of the FBE model. Furthermore, a parametric study conducted on the last case study is presented.

Chapter 6 - Conclusions and recommendations
The final chapter summarises the important findings of this research. Recommendations for future research and development in this field are discussed.

Chapter 2

Literature review

2.1 Introduction

The aim of this chapter is to provide information from literature that is required to understand the remaining chapters. Firstly, the fields of fire safety and structural fire engineering are introduced. Secondly, an explanation of fire models and the difference between real fires and fire curves used in practice is provided. Thereafter, an outline of how building materials respond to fire is presented, including a brief summary of the Eurocode material models of steel and concrete in fire. This is followed by an explanation of important concepts in the analysis of structures in fire. Lastly, the available software used for such analyses is presented.

2.2 Fire safety

Fires in buildings cause thousands of deaths and billions of dollars' worth of destruction each year (Buchanan and Abu, 2017). In the last two decades, the breakout of fires in and subsequent collapse of buildings has attracted a great deal of publicity and concern. The re-evaluation of fire safety was triggered by the catastrophic collapse of the World Trade Center (WTC) in New York on September 11, 2001 (NIST, 2005), especially WTC7 in which mechanical damage played a very minor role in the fire-induced collapse. More recent disasters such as the Grenfell Tower fire in London in May 2017 (Gorse and Sturges, 2017), again emphasized the importance of fire resistance and the responsibility of engineers and designers to seriously consider fire safety in buildings. Buchanan and Abu (2017) state that "the primary goal of fire protection is to limit, to acceptable levels, the probability of death, injury, property loss and environmental damage in an unwanted fire." With this in mind, humankind continues to learn and develop practices to minimise the risk and mitigate the dangers of fire (Wang *et al.*, 2012).

2.3 Structural fire engineering

An important branch of fire safety is structural fire engineering, a relatively new discipline that aims to predict the behaviour of buildings in fire and to design structures accordingly (Lennon, 2011). The field has contributed to overall building fire

safety by providing fire resistance in structural elements (Buchanan and Abu, 2017). This may be achieved through prescriptive or performance-based design and by the application of active or passive protection. These terms are discussed in more detail in the sections that follow.

2.3.1 Prescriptive versus performance-based design

Currently, there are two main approaches to structural design for fire: prescriptive and performance-based. The prescriptive approach is more traditional while performance-based design methods were only developed staring in the early 1970s (Duthinh, 2014).

Prescriptive methods are a set of rules that can be implemented with ease by a designer. They typically include fixed values, safety factors and a performance rating in a standard fire (refer to Section 2.4.2). Examples include tabulated data sets giving specified intumescent paint thicknesses, compartmentation sizes, sprinkler requirements or concrete reinforcement cover depths to suit different building requirements. However, the engineering principles upon which they are based are sometimes vague. The main disadvantages are that the solutions are typical uneconomical and designers simply apply rules without having a sound understanding of how the structure would behave in the case of a real fire (Wang *et al.*, 2012).

To address these disadvantages, performance-based design is being adopted by designers. Performance-based design includes the determination of realistic fire scenarios, from which solutions based on the expected structural response can be developed. These innovative methods are less conservative and therefore more economical, but demand a greater skill and understanding of structural performance in fire.

In recent years, considerable progress has been made with regards to performance-base design, however, many of the modelling techniques developed are often too complex to be commonly utilised by designers in practice. As already explained in Chapter 1, applying these principles in performance-based designs would demand too much computational power and time-cost for most consulting engineers. Also, very little technical guidance is available for such designs. There is therefore a need to develop simple analytical expressions that are easy to apply, while still being correct and scientific (Usmani *et al.*, 2001).

2.3.2 Active and passive protection

The control of a fire in a buildings can be achieved by active or passive protection. Active protection includes any control measure that is triggered in the event of a fire, for example, an automatic sprinkler system and fire and rescue personal (Buchanan and Abu, 2017). Active protection is effective for smaller fires and, if it is not applied in the early stage of the fire's development, will not reduce the structural fire severity of a fully developed fire.

In contrast to the above, passive protection is the inherent fire resistance that is provided by the structure or material of the building. It does not require an activation but is rather included in the design and development phase of a structure. Examples of passive control include materials that prevent the spread of fire, such as a fire wall or fire resistant structural assemblies (Buchanan and Abu, 2017).

2.4 Fire models

Fire dynamics is a wide field in itself and structural fire engineers require a fundamental knowledge of fire behaviour to be able to successfully design fire resistant structures. Most importantly, designers must have a sound understanding of the difference between real fires and the use of benchmark fire curves. For a detailed discussion on fire dynamics and behaviour refer to Quintiere (2017).

2.4.1 Real fires

A real fire is a complex phenomenon with different phases and is influenced by a range of factors. A typical time-temperature behaviour of a real fire is depicted in Figure 2.1. The growth of a real fire typically begins gradually, known as the smouldering phase, and then rapidly accelerates until flash-over occurs, which is a state of full-room involvement (Thomas *et al.*, 1980). The maximum temperature of real fires is known to be as high as 1200°C. The post-flashover fire phase is considered the most critical part for designers (Buchanan and Abu, 2017). Thereafter the fire enters a decay phase as the fuel gets burned up. Real fires in large open spaces also behave as travelling fires (Stern-Gottfried and Rein, 2012). These fires spread across the floor burning over a specific area at any one time.

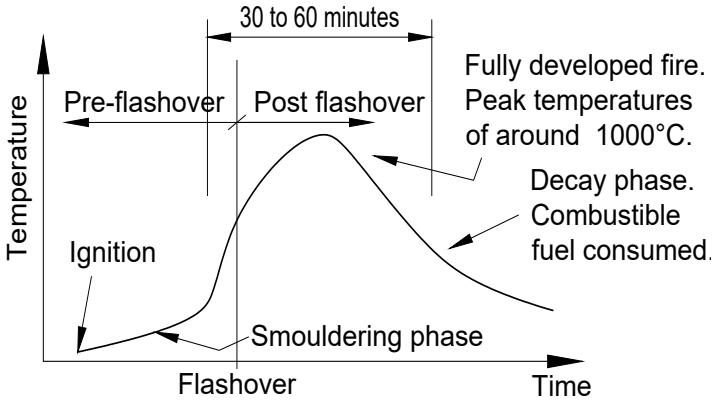


Figure 2.1: Time-temperature behaviour of a real fire (Walls, 2016)

2.4.2 Design fires

In reality, no fire is the same as another, but structural engineers require a design fire in order to consistently measure a structures' fire resistance rating (FRR). These

design fires are theoretical mathematical models used as benchmarks.

The ISO 834 (ISO, 1999) standard fire is the most commonly used benchmark fire in structural fire design. Despite having received criticism as being unrealistic and unscientific (Law, 1981), this standard fire is used in furnace tests worldwide to determine the FRR of materials or structural elements. The FRR is a duration of time in increments of 30 minutes, where a 30 min FRR is low whereas a 2 hour FRR is considered high. This theoretical fire curve is defined by Equation (2.4.1), where t is the time in minutes and θ_g is the gas temperature in °C.

$$\theta_g = 20 + 345 \log_{10}(8t + 1) \tag{2.4.1}$$

Similarly, the benchmark used for hydrocarbon fires in the petrochemical industry is defined by the following curve:

$$\theta_g = 1080(1 - 0.325e^{-0.167t} - 0.675e^{-2.5t}) + 20 \tag{2.4.2}$$

A less intense fire curve can be used for external fires:

$$\theta_g = 660(1 - 0.687e^{-0.32t} - 0.313e^{-3.8t}) + 20 \tag{2.4.3}$$

The time-temperature representation of these benchmark fires can be seen in Figure 2.2.

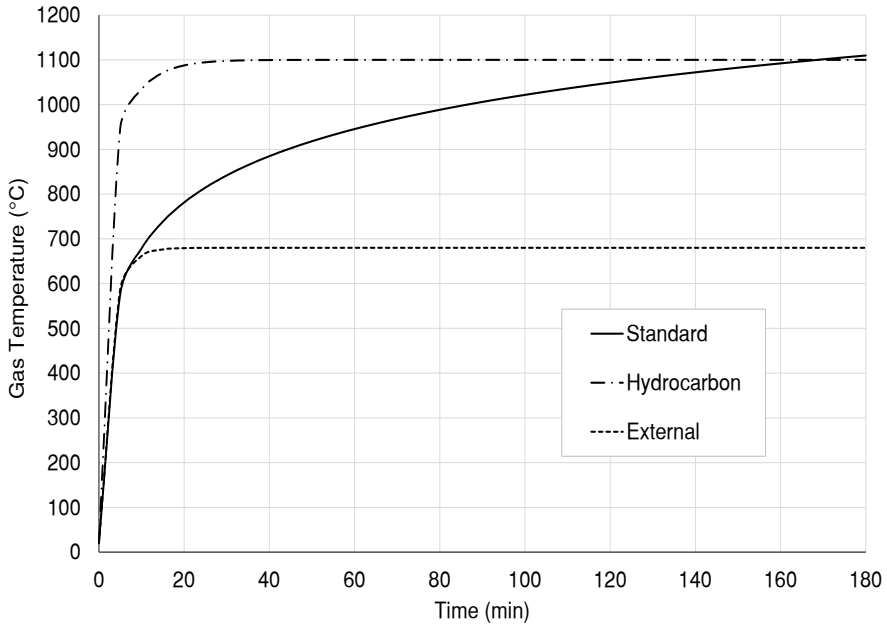


Figure 2.2: Time-temperature fire curves

2.5 Material models in fire

Material response to high temperature is an important part of performance-based design. Both steel and concrete show a progressive reduction in stiffness and strength as temperature increases, even in temperatures considerably below the material's melting point (Wang *et al.*, 2012). These thermo-mechanical properties of building materials need to be specified as input parameters for analyses and are therefore discussed in this section.

The most recent research is captured in the Eurocodes by providing formulae and curves for the thermal response of construction materials (Wang *et al.*, 2012). For this reason, the material models in the Eurocode EN 1993-1-2 (BSI, 2005) for steel and EN 1992-1-2 (BSI, 2004*a*) for concrete have been adopted throughout this research. The most important thermal properties of construction materials applicable in design, and therefore this research, are thermal expansion, stress-strain response, and the strength and stiffness reduction factors, and will be presented in the following sections.

2.5.1 Steel in fire

Unprotected steel is vulnerable in fire as a result of its high thermal conductivity and thin members. Increases in temperature of a steel structure can lead to significant deformation and failure. At 750°C the steelwork undergoes a phase change and drastically loses its strength and stiffness. However, physical melting only occurs at temperatures well over 1500°C (Wang *et al.*, 2012).

2.5.1.1 Thermal expansion of steel

Curves for thermal strain (ε_θ) for various typical carbon steels, including reinforcing steels, are provided in the Eurocode EN 1993-2-1 (BSI, 2005) as depicted in Figure 2.3. These are dependent on the steel temperature (θ_a), as can be seen by the following equations:

$$\varepsilon_\theta = -2.416 \times 10^{-5} + 1.2 \times 10^{-5}\theta_a + 0.4 \times 10^{-8}\theta_a^2 \qquad \text{for } 20°\text{C} < \theta_a \leq 750°\text{C} \tag{2.5.1}$$

$$\varepsilon_\theta = 11 \times 10^{-3} \qquad \text{for } 750°\text{C} < \theta_a \leq 860°\text{C} \tag{2.5.2}$$

$$\varepsilon_\theta = -6.2 \times 10^{-3} + 2 \times 10^{-5}\theta_a \qquad \text{for } 860°\text{C} < \theta_a \leq 1200°\text{C} \tag{2.5.3}$$

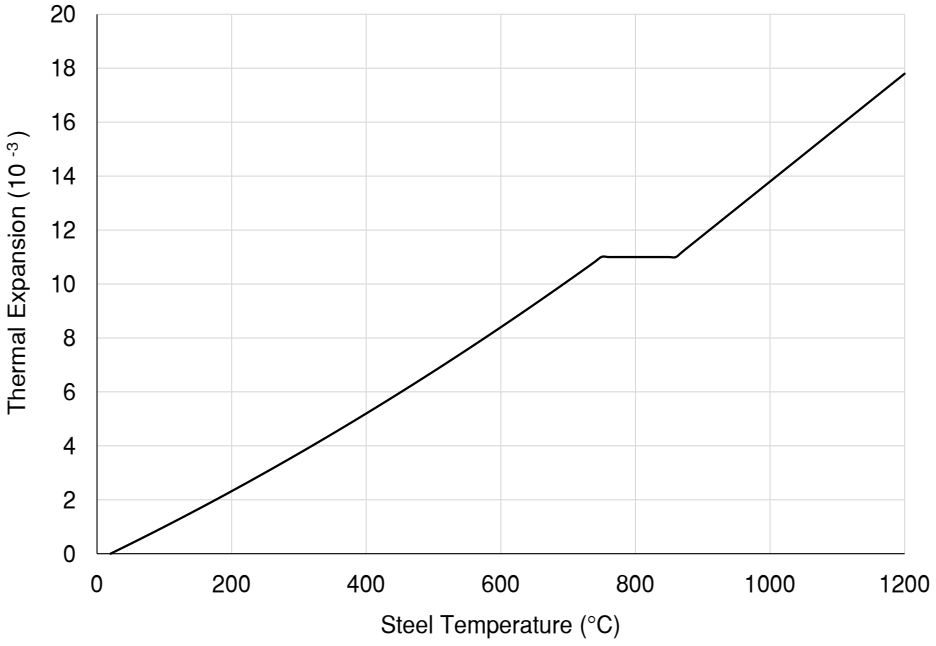


Figure 2.3: Steel thermal expansion as a function of temperature (BSI, 2005)

2.5.1.2 Mechanical properties of steel at high temperature

For conventional design, steel is considered to have a very specific yield strength f_y, but in fire design, this is no longer valid because the stress-strain response of steel changes as temperature increases. The reduction of strength and stiffness in steel at elevated temperatures is significant (Wang *et al.*, 2012) and the Eurocode accounts for this by applying reduction factors. These factors are temperature-dependent and are multiplied by the original strength or stiffness at 20°C. They are defined as follows:

- effective yield strength reduction factor: $k_{y,\theta} = f_{y,\theta}/f_y$ (2.5.4)
- proportional limit reduction factor: $k_{p,\theta} = f_{p,\theta}/f_y$ (2.5.5)
- slope of linear elastic range reduction factor: $k_{E,\theta} = E_\theta/E$ (2.5.6)

The values for reduction factors can be read off Figure 2.4.

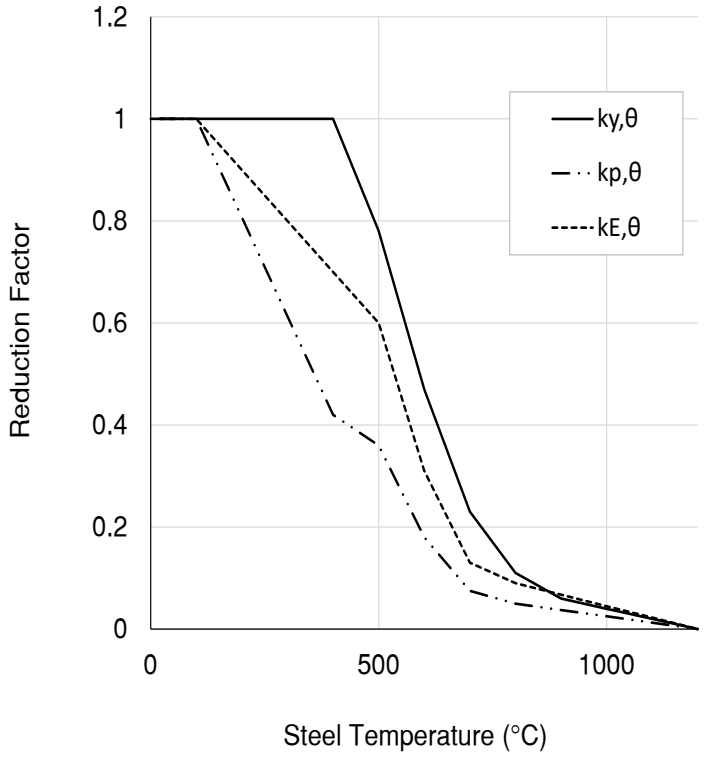


Figure 2.4: Steel reduction factors as a function of temperature (BSI, 2005)

Eurocode EN 1993-1-2 specifies high-temperature stress-strain curves for a range of temperatures. The curve in Figure 2.5 shows the typical behaviour of carbon steel when strain increases:

1. At first, there is a linear range form zero to the limit of proportionality $f_{p,\theta}$ by a gradient equal to the value of the Young's Modulus E_θ, which specific to that temperature.

2. After the limit of proportionality, the material is in a transition phase until it yields at its yield strain $\varepsilon_{y,\theta}$.

3. Thereafter, a stress plateau at $f_{y,\theta}$ is maintained until the limiting strain for yield strength, $\varepsilon_{t,\theta}$, is reached.

4. The final fracture process is represented by the negative slope up until the ultimate strain, $\varepsilon_{u,\theta}$, of the material. (Wang *et al.*, 2012)

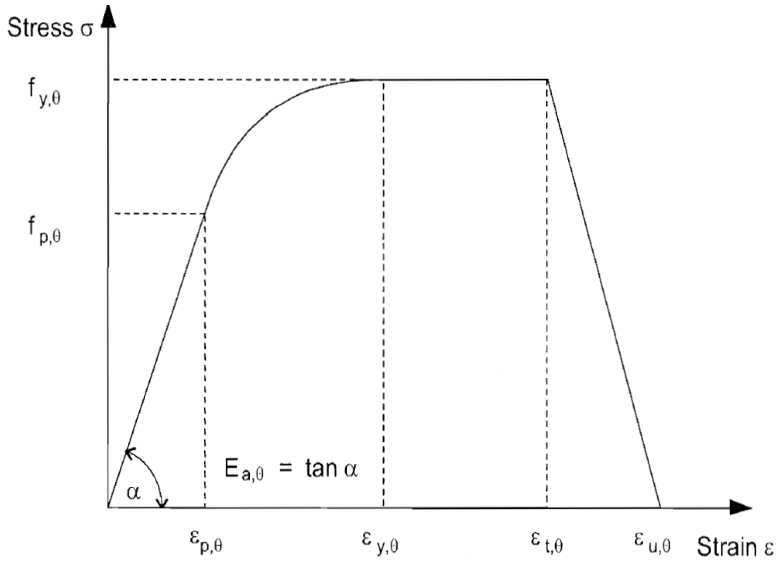


Figure 2.5: Steel stress-strain curve (BSI, 2005)

The stress-strain curves at different temperatures all show a similar shape, but the higher the temperature, the lower the stress pattern. Note that these curves encompass the effects of creep on yield stress (Wang *et al.*, 2012), but strain hardening has been conservatively ignored. EN 1993-1-2 (BSI, 2005) provides the following equations to represent the curves:

$$\sigma = \varepsilon E_\theta \qquad for\ 0 < \varepsilon \leq \varepsilon_{p,\theta} \tag{2.5.7}$$

$$\sigma = f_{p,\theta} - c + \frac{b}{a}\sqrt{a^2 - (0.02 - \varepsilon)^2} \qquad for\ \varepsilon_{p,\theta} < \varepsilon \leq \varepsilon_{y,\theta} = 0.02 \tag{2.5.8}$$

$$\sigma = f_{y,\theta} \qquad for\ \varepsilon_{y,\theta} < \varepsilon \leq \varepsilon_{t,\theta} = 0.15 \tag{2.5.9}$$

$$\sigma = f_{y,\theta}(1 - \frac{\varepsilon - 0.15}{0.05}) \qquad for\ \varepsilon_{t,\theta} < \varepsilon \leq \varepsilon_{u,\theta} = 0.20 \tag{2.5.10}$$

$$\sigma = 0.0 \qquad for\ \varepsilon > \varepsilon_{u,\theta} \tag{2.5.11}$$

The term c is calculated by:

$$c = \frac{(f_{y,\theta} - f_{p,\theta})^2}{(\varepsilon_{y,\theta} - \varepsilon_{p,\theta})E_\theta - 2(f_{y,\theta} - f_{p,\theta})} \tag{2.5.12}$$

This is then substituted to determine a and b:

$$a^2 = (0.02 - \varepsilon_{p,\theta})(0.02 - \varepsilon_{p,\theta} + \frac{c}{E_\theta}) \tag{2.5.13}$$

$$b^2 = c(0.02 - \varepsilon_{p,\theta})E_\theta + c^2 \tag{2.5.14}$$

2.5.1.3 Reinforcing steel in fire

EN 1992-1-2 specifies that the strength and deformation of reinforcing steel in fire can be calculated with the same models as provided for structural steel as specified in the previous sections. Note that throughout this document, strain hardening has conservatively been neglected as recommended by EN 1992-1-2.

2.5.2 Concrete in fire

Concrete is generally considered to perform better in fire than steel, as it has a low thermal conductivity (Buchanan and Abu, 2017). However, the behaviour of concrete in fire is significantly more complex. A variety of parameters influence the characteristics of concrete as it is made of a matrix of different materials. As temperature increases, the materials are subjected to chemical and physical changes. Melting starts at around 1200°C and between 1300°C and 1400°C the concrete has melted completely (Wang *et al.*, 2012). In this research, the concrete model from Eurocode EN 1992-1-1 (BSI, 2004*a*) and EN 1992-1-2 (BSI, 2004*b*) has been adopted.

2.5.2.1 Thermal expansion of concrete

The thermal expansion of concrete varies depending on the aggregate type (Wang *et al.*, 2012) as its composition and crystal structure influences the thermal strains. For this reason, the Eurocode differentiates between siliceous- and calcareous-aggregate concretes, as can be seen in Figure 2.6.

The thermal expansion for siliceous-aggregate concrete can be calculated as follows:

$$\varepsilon_\theta = -1.8 \times 10^{-4} + 9 \times 10^{-6}\theta_c + 2.3 \times 10^{-11}\theta_c^3 \qquad for\ 20°\text{C} < \theta_c \leq 700°\text{C} \tag{2.5.15}$$

$$\varepsilon_\theta = 1.4 \times 10^{-2} \qquad for\ 700°\text{C} < \theta_c \leq 1200°\text{C} \tag{2.5.16}$$

The thermal expansion for calcareous-aggregate concrete is given by:

$$\varepsilon_\theta = -1.2 \times 10^{-4} + 6 \times 10^{-6}\theta_c + 1.4 \times 10^{-11}\theta_c^3 \qquad for\ 20°\text{C} < \theta_c \leq 805°\text{C} \tag{2.5.17}$$

$$\varepsilon_\theta = 1.4 \times 10^{-2} \qquad for\ 805°\text{C} < \theta_c \leq 1200°\text{C} \tag{2.5.18}$$

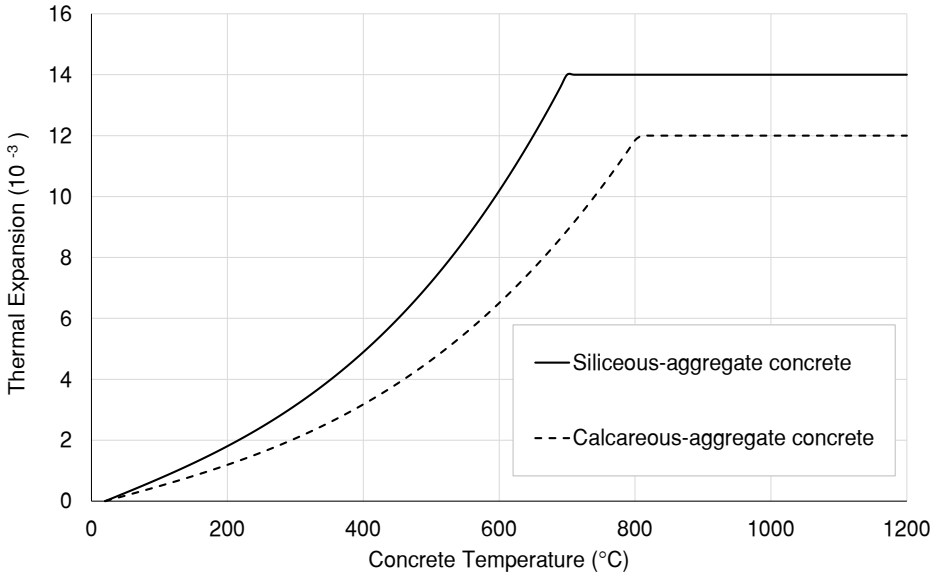


Figure 2.6: Concrete thermal expansion as a function of temperature and aggregate content (BSI, 2004*b*)

2.5.2.2 Mechanical properties of concrete in compression

Conventional stress-strain curves display a significant loss of compressive strength ($f_{c,\theta}$) and change in stress-strain response as temperature increases. The typical stress-strain relationship of concrete has been simplified in the code into the curve in Figure 2.7. The equation for the curve are provided as follows:

$$\sigma = \frac{3\varepsilon f_{c,\theta}}{\varepsilon_{c1,\theta}(2 + (\frac{\varepsilon}{\varepsilon_{c1\theta}})^3)} \quad for\ \varepsilon \leq \varepsilon_{c1,\theta} \tag{2.5.19}$$

linear or non-linear models are permitted $for\ \varepsilon_{c1,\theta} < \varepsilon \leq \varepsilon_{cu1,\theta}$

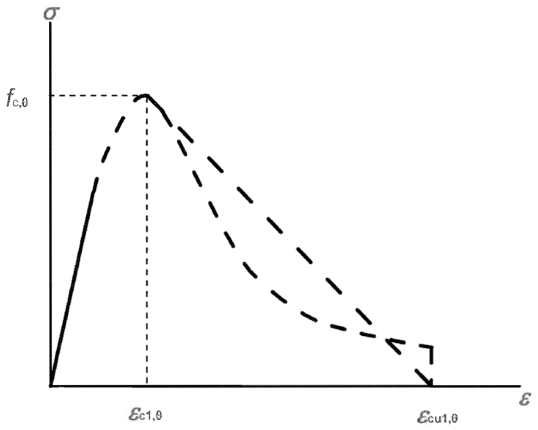


Figure 2.7: Concrete stress-strain curve (BSI, 2004*b*)

EN 1992-1-2 specifies the following parameters for both siliceous- and calcareous-aggregate concrete:

- Compressive strength reduction factor: $f_{c,\theta}/f_{ck}$
- Concrete strain at maximum stress: $\varepsilon_{c1,\theta}$
- Concrete strain at failure: $\varepsilon_{cu1,\theta}$

These values are provided in Table 2.1 for a range of temperatures.

Table 2.1: Values for the main parameters of the stress-strain relationships of normal weight concrete at elevated temperatures (BSI, 2004*b*)

Concrete Temperature θ_c	**Siliceous Aggregates**			**Calcareous Aggregates**		
(°C)	$f_{c,\theta}/f_{ck}$	$\varepsilon_{c1,\theta}$	$\varepsilon_{cu1,\theta}$	$f_{c,\theta}/f_{ck}$	$\varepsilon_{c1,\theta}$	$\varepsilon_{cu1,\theta}$
20	1.00	0.0025	0.0200	1.00	0.0025	0.0200
100	1.00	0.0040	0.0225	1.00	0.0040	0.0225
200	0.95	0.0055	0.2500	0.97	0.0055	0.0250
300	0.85	0.0070	0.2750	0.91	0.0070	0.0275
400	0.75	0.0100	0.0300	0.85	0.0100	0.0300
500	0.60	0.0150	0.0325	0.74	0.0150	0.0325
600	0.45	0.0250	0.0350	0.60	0.0250	0.0350
700	0.30	0.0250	0.0375	0.43	0.0250	0.0375
800	0.15	0.0250	0.0400	0.27	0.0250	0.0400
900	0.08	0.0250	0.0425	0.15	0.0250	0.0425
1000	0.04	0.0250	0.0450	0.06	0.0250	0.0450
1100	0.01	0.0250	0.0475	0.02	0.0250	0.0475
1200	0.00	-	-	0.00	-	-

2.5.2.3 Mechanical properties of concrete in tension

As concrete is a brittle material, tensile strength in concrete cannot easily be guaranteed. Even in ambient design, tensile strength in concrete is usually assumed to be zero. However, in whole-structural analyses the designer can consider some of the tensile strength according to the guidelines of EN 1992-1-2 (BSI, 2004*b*).

At elevated temperatures, the tensile strength of concrete also reduces and therefore BSI (2004*b*) again applies a reduction factor for strength ($k_{c,t}$). This factor simply follows a linear reduction in strength between 100°C and 600°C. After 600°C the concrete is assumed to have lost all strength in tension. The factor can be calculated as follows:

$$k_{c,t} = 1.0 \qquad \text{for } 20°\text{C} \leq \theta_c \leq 100°\text{C} \tag{2.5.20}$$

$$k_{c,t} = 1.0 - \frac{\theta_c - 100}{500} \qquad \text{for } 100°\text{C} < \theta_c \leq 600°\text{C} \tag{2.5.21}$$

Walls (2016) dedicated a section in his dissertation to explain how tension stiffening models could be incorporated in analysis of beam elements in fire. However, due to the complexity of these models, tension stiffening has been conservatively neglected throughout this document.

2.6 Analysis of structures in fire

Structures behave very differently in fire conditions than they would at ambient temperature. There is a significant change in load paths, stresses, material behaviour and deflection when structures are exposed to high temperature. This means that designers require a sound understanding of global structural behaviour at the fire limit state (Buchanan and Abu, 2017). This section provides background on fire design as opposed to ambient design. The difference between a coupled and a decoupled analysis is explained. A brief note on the importance of global analysis in fire is followed by an explanation of structural behaviour in restrained structures, which is the focus of this research.

2.6.1 Fire design versus ambient design

For ambient design, a great number of assumptions made regarding loading, material and design requirements, allow for simplifications in design methods. However, at the fire limit state, these assumptions are no longer valid (Gillie, 2009):

1. Ambient design accounts for static loading, while a fire is a variable load. The stresses during a fire are not constant.

2. As discussed in Section 2.5, most building materials experience a general loss of linearity, strength, Young's Modulus and a clear yield point at higher temperatures.

3. Ambient design assumes small deflections. This no longer applies to fire design, as thermal expansion may cause large deflection and buckling, as seen in Figure 2.8.

Figure 2.8: Beams after Cardington fire test (Lamont, 2001)

4. An important structural response in fire design is thermal bowing, which is the effect when the exposed side of a member heats up significantly more than the other side. The heated side will try to expand resulting in an induced curvature (Usmani *et al.*, 2001). This is typically seen in composite members, where the steel would reach higher temperatures than the concrete slab.

As a result of all these factors, fire design has to account for a complex combination of time and temperature-varying response with material and geometric non-linearity (Gillie, 2009).

2.6.2 Decoupled vs coupled analysis

The structural analysis of a building in fire is dependent on the heat transfer analysis. These two processes can either be decoupled or coupled, depending on the level of accuracy required and input data available (Wang *et al.*, 2012).

A decoupled analysis begins with the heat transfer analysis to determine the temperatures of the structure for the duration of the fire. These temperatures are then used as input data for the mechanical analysis, which is performed separately. An example of a decoupled analysis is the work of Marx (2018) and Kloos (2017) on a novel cellular beam structure in fire. Marx (2018) performed the heat transfer analysis, which was used in the work of Kloos (2017) in the structural analysis. Similarly, the work done in this research adopts the decoupled analysis approach. The method focusses on the second part of the decoupled analysis, namely the structural response, and does not include heat transfer analyses. Instead, reliable temperature data sets were used as input.

A fully coupled analysis, in comparison, is far more complex, as heating and mechanical response has to be modelled simultaneously and finite elements (FE) require thermal and mechanical degrees of freedom (DOFs). The coupled approach requires considerably more computing time and power, and is therefore rarely used (Wang *et al.*, 2012). It is only necessary when the mechanical response directly influences the subsequent thermal response, for example in steel decks when debonding from concrete slabs occurs.

2.6.3 Local vs global analyses

Analysis of structures in fire can either be performed using a local or a global model. Local analyses consider the structural elements in isolation and with idealised support conditions (Wang *et al.*, 2012), while global analyses incorporate the whole structure or substructure with appropriate boundary conditions (for example symmetric or spring boundary conditions). To illustrate the difference, an example from the work of Kloos (2017) on the analysis of a cellular beam structure is provided. Figure 2.9 shows the finite element analysis (FEA) performed in ABAQUS (Dassault Systeme Simulia, 2016) of an isolated beam, whereas Figure 2.10 shows the global analysis of the beams in a floor structure.

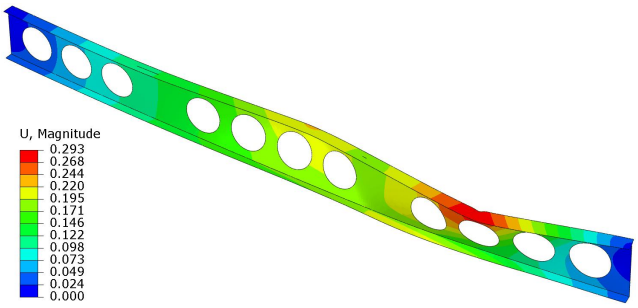


Figure 2.9: Local analysis of a cellular beam by Kloos (2017)

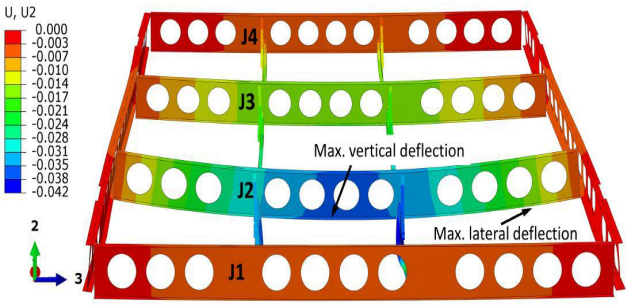


Figure 2.10: Global analysis of a cellular beam floor structure by Kloos (2017)

Gillie (2009) emphasises that for fire analysis, accurate results may only be obtained if whole structures or parts of structures are analysed globally. Work by Buchanan and Abu (2017) and Usmani *et al.* (2001) also highlight the importance of the analysis of global structures in fire as opposed to individual elements. Continuity and axial restraint have a considerable influence on structural behaviour which cannot be captured in individual element testing methods. For example, the effect of elongations and rotations on cooler areas of the building require consideration. For more information on different types of global analyses refer to Wang *et al.* (2012).

2.6.4 Modelling structures with continuity

Fire creates a complicated distribution of forces when considering the global structure. The relationship between members, temperature gradients, temperature differences, thermal expansions, material degradation and 3D behaviour have to be considered. An element in a compartment that is exposed to fire, for example, is affected by the elements outside of the compartment, as they are cooler and stiffer (Sanad *et al.*, 2000).

2.6.4.1 Thermal strains and restraint

In common practice, fire resistance is based on isolated elements in furnaces. However, this analysis has been considered "over-conservative and even more importantly,

unscientific" by Usmani *et al.* (2001). They further state that the most important factor that influences the behaviour of a structure under fire is the response to thermal strains. A structure with limited restraint will give a deflection-dominated response as the thermal strains will induce lengthening and curvature. In contrast, with restraint present, the deflections will be less but internal forces may be higher. This is as a result of the large compressive mechanical strains induced in the restrained member. Rotational restraint leads to large hogging moments while translational restraint only leads to large tensile forces. Furthermore, the mechanical strains that develop may exceed the yield stress resulting in extensive plastification. The result is a member with lower deflection but reduced stiffness. In summary, the two fundamental responses of restrained beams are yielding and buckling, or a complex combination of the two (Usmani *et al.*, 2001). Yielding typically occurs in more stocky members while slender members are more prone to buckling.

Most models assume perfectly rigid connections with infinite restraint. In real structures however, perfectly rigid connections do not exist. The restraint is finite and can be represented by spring stiffnesses. For more information regarding restraint in structures refer to the work of Usmani *et al.* (2001).

2.6.4.2 The effect of axial restraint

Axial restraint plays an important role in reinforced or pre-stressed concrete slabs or beams, as the axial force compensates for the weakened steel reinforcing. Buchanan and Abu (2017) provide an explanation of the effect of restraint on heated members by heating the bottom of a simply supported concrete beam between rigid supports, as depicted in Figure 2.11.

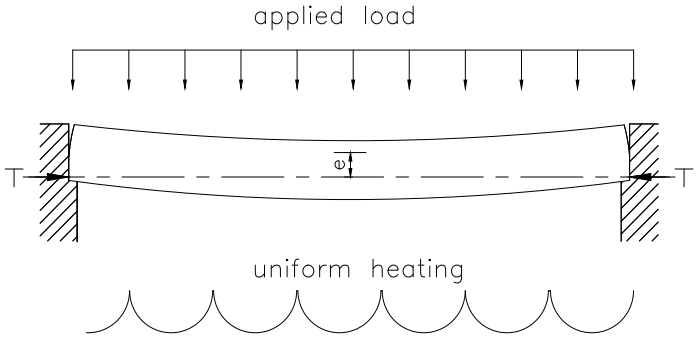


Figure 2.11: Heated beam with axial restraint (recreated from Buchanan and Abu (2017))

The beam tries to expand but the rigid supports create a thrust force T as a reaction at an eccentricity e to the middle of the compression block. This thrust force contributes to the members strength and flexural resistance by inducing a couple

Te. The couple moment would typically act against the applied moment of the load and thereby resist the load at fire limit state. Conversely, axial restraint may also have the opposite effect, if the thermally induced thrust develops near the top of the section. Also, if deflections and rotations become excessive, axial restraint becomes difficult to predict (Buchanan and Abu, 2017).

2.6.5 Fire testing

Fire testing plays an important role in the structural fire design field. It provides insight into structural behaviour in fire as well as being useful for the calibration of computer models (Bisby *et al.*, 2013).

There are currently two approaches to fire testing. Firstly, the standard furnace tests which use the ISO 834 (ISO, 1999) fire curve. These tests typically include isolated elements with simplified boundary conditions (fully-fixed or pinned) in a furnace (Wang *et al.*, 2012). However, in recent years these tests have been criticised by various authors including Usmani *et al.* (2001) and Bisby *et al.* (2013) as not being representative of the true behaviour of structures in fire. There are various effects as a result of continuity, restraint, redistribution and member action that cannot be captured in the standard fire test scenario. Nevertheless, Bisby *et al.* (2013) agree that standard furnace testing has provided a useful tool for comparison and benchmarking.

The second type of testing is full-scale fire tests, which are assemblies or sub-assemblies that are exposed to a fire of an equivalent fuel load. These tests are performed with the aim of understanding complex interactions of structural mechanisms in a real fire (Usmani *et al.*, 2001). Well-known examples of full-scale tests are the Cardington Steel Building Tests performed in 1993 to 2003, which have been regarded as a breakthrough in structural fire engineering. Seven tests were performed on steel frames and the results were used for research (for example by Franssen *et al.* (1995)), calibration (for example, the benchmarks in COST (2014)) and design codes and guidelines. For more details regarding the Cardington fire tests refer to the Cardington Main Report (Usmani *et al.*, 2000).

2.7 Software in structural fire engineering

Fire analyses have an inherent high complexity and non-linear behaviour (as explained in Section 2.6.1) and therefore numerical models applying FEA are widely accepted as the most appropriate tool for structural fire analyses (Wang *et al.*, 2012). The software specifically designed for fire engineering is typically researched-based, although some commercial software have included a fire analysis module.

Commercial software is useful in fire analysis as it typically provides a large range of element types, solution algorithms and various applications (i.e. it is not limited to only fire analyses). Examples of these FEA packages that have been used for fire applications are ABAQUS (Dassault Systeme Simulia, 2016), Ansys (2016), DIANA

(TNO DIANA, 2016) and LS-DYNA (LSTC, 2014). These packages usually follow the decoupled approach to fire analysis. ABAQUS is a powerful and versatile FEA software package that has the option of linking thermal analyses to subroutines. LS-DYNA is a general-purpose finite element software especially developed for highly non-linear and transient dynamic analysis and has shown considerable growth in terms of the use in fire engineering. The disadvantage of commercial software is the high cost and the limitations of the analyst in terms of accessing and modifying the code (Wang *et al.*, 2012).

Research-based fire software is specifically developed for detailed fire analyses and example of such include SAFIR (Franssen, 2005), CEFICOSS, Vulcan (Burgess, 2015) and OpenSees (Jiang and Usmani, 2013). SAFIR and its predecessor, CEFICOSS were designed especially for thermal and structural analyses of structures in fire and have been developed at Liege University, Belgium. Vulcan is more specific for fire analysis of composite structures and developed at the University of Sheffield. OpenSees (McKenna, 1997) is a open-sourced FEA software developed by the University of California, Berkeley, for earthquake analysis. However, the University of Edinburgh added a fire application to the software (Jiang and Usmani, 2013). OpenSees for fire will further be discussed in detail in Chapter 4. Vulcan and OpenSees both use a combination of shell elements for the slabs and beam elements for composite structures.

2.8 Summary

This chapter explained the role of structural fire engineering in ensuring fire safety in buildings. The analysis tools available for designers are fire models and material models as specified in the Eurocodes EN 1993-2-1 and EN 1992-1-2. Furthermore, the complexities of fire design, as opposed to ambient design, were explained. Fire analysis concepts were provided as background. This included a brief summary of fire testing methods, as well as FEA software available for fire design.

Chapter 3

Theory of the Fire Beam Element methodology

3.1 Introduction

This chapter discusses the main principles of the Fire Beam Element (FBE) method for the analysis of structures in fire. The FBE concept is introduced by providing a brief background on its development. Then the need of a fibre section is discussed. Thereafter, the structural mechanics of stresses and strains in fire are outlined. Building on this concept, the calculation of the section stiffness and neutral axis (NA) position of a beam element is explained, and illustrated by a simple example. This is followed by an explanation of how FBE is integrated into FEA theory. Furthermore, the design steps of the FBE method applied to a 3D composite structure is outlined. Lastly, advantages and limitations of the FBE formulation are discussed.

The work presented in this chapter is based primarily upon the original FBE work (Walls, 2016; Walls *et al.*, 2018), but is provided as it is essential for understanding the structural mechanics employed. This work will be developed upon in the following chapters. Unless noted otherwise, the details below are based upon the aforementioned first version of the FBE work.

3.1.1 Background to the fire beam element

The concept of using beam elements with NA positions that vary for the analysis of structures in fire first appeared in literature in Bresler *et al.* (1977). The authors set up a programme to analyse concrete frames in fire. Frame elements were discretised by beam elements with linear moment distribution along the beam axis. Bresler *et al.* (1977) further noted that the analysis was especially complex with constantly varying internal forces, as a result of restraint on free thermal expansion, shrinkage or creep.

As discussed in Chapter 1, the initial development work on the FBE was carried out by Walls (2016). This method has the aim to provide an analysis tool that is simpler

than the currently-used FEM methods, but accurate enough to simulate structural behaviour in real fires. In essence, the method divides the structure into a skeletal beam model with shifting neutral axes. By validation examples, Walls demonstrated that the method yielded sufficiently accurate results in comparison to experimental results and contemporary structural analysis systems. However, the work was only applied in a simplified manner, and to structures without restraint. Before the FBE method can be utilised by designers and consultants, further research in the application of the FBE method to fully restrained or partially-restrained structures was recommended. This recommendation has been the focus of this research.

3.2 The fibre section

In a fire scenario, the cross-section of a beam may be exposed to a thermal gradient. To model this non-uniform temperature distribution, the fibre element approach is applied to the cross-sections. The height of the section is discretised into n number of smaller rectangular fibres, as seen is Figure 3.1. Each fibre i represents a small rectangular area A_i of the section and a second moment of inertia I_i about the NA. Note that this I_i value changes as the NA shifts.

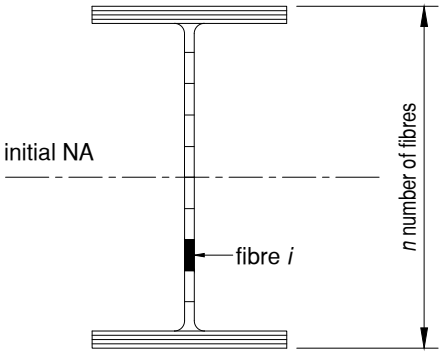


Figure 3.1: Example of a steel profile being discretised into n number of fibres

The fibre section approach is useful for FBE for the following reasons:

- A different temperature can be assigned to each fibre, thereby simulating a thermal gradient.
- The fibre section caters for composite sections as each fibre is assigned a material model.
- The fibre analogy is further useful in capturing gradual yielding and plasticity distribution over a section in fire.

The fibre element approach has been used in the field of structural fire modelling in work by Franssen *et al.* (1995), Burgess *et al.* (2000), Bailey (1998) and Jeffers and Sotelino (2012). The difference between the aforementioned research and this

work include (a) the manner in which the NA is updated, (b) the consideration of composite sections, and (c) the calculation and application of thermal forces. How this is done will be explained in detail in the remaining sections of this chapter.

3.3 Strains in fire

The total strain ε according to EN 1992-1-2 (BSI, 2004*b*) is the sum of thermal strain (ε_θ), mechanical strain (ε_σ), creep strain (ε_{creep}) and transient strain (ε_{tr}) as displayed in Equation (3.3.1).

$$\varepsilon = \varepsilon_\theta + \varepsilon_\sigma + \varepsilon_{creep} + \varepsilon_{tr} \tag{3.3.1}$$

However, the Eurocode implicitly incorporates creep and transient strain in the material models and reduction factors (see Section 2.5). This simplifies Equation (3.3.1) to Equation (3.3.2) and is illustrated by Figure 3.2.

$$\varepsilon = \varepsilon_\theta + \varepsilon_\sigma \tag{3.3.2}$$

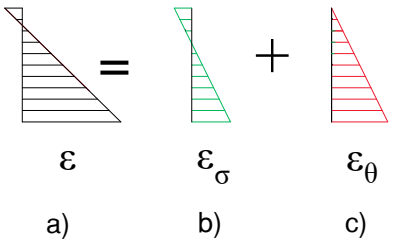


Figure 3.2: Example of a) the total strain in a cross-section as the combination of b) the mechanical strain and c) the thermal strain. The thermal load in this example has a uniform temperature gradient over the depth of the cross-section.

The thermal strain develops as a response to a temperature increase and can be calculated directly according to the material models specified. In this research, the thermal strains are calculated according to the Eurocodes (BSI, 2005, 2004*b*) as explained in Section 2.5.1.1 for steel and in Section 2.5.2.1 for concrete. The calculation of the total and mechanical strain will be explained in Section 3.5.2.1. Note that the sign convention adopted in this research for strains and stresses is taken as negative for compression and positive for tension.

3.4 Forces and stresses in fire

This section provides a brief outline of the fundamental principles of structural behaviour at high temperatures. For more detail on this topic, the reader is referred to Usmani *et al.* (2001).

First consider two beams of length L where a) is a cantilever beam while beam b) is fully restrained on both ends as depicted in Figure 3.3. Both beams are uniformly heated with a temperature difference of ΔT and no other loading is applied.

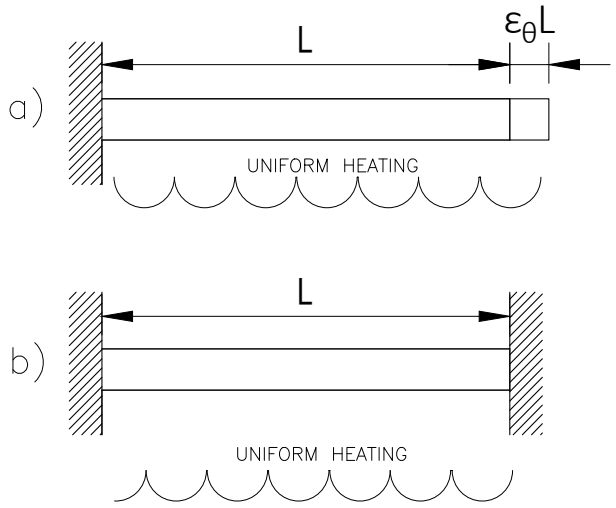


Figure 3.3: Uniform heating of a) a cantilever beam and b) a fully restrained beam

Beam a) is free to elongate by $\varepsilon_\theta L$ where the thermal strain ε_θ is calculated according to the Eurocodes (BSI, 2005, 2004*b*) (refer to Section 2.5). Since no other load is applied, ε_σ is zero and the total strain will equal the thermal strain:

$$\varepsilon = \varepsilon_\theta \tag{3.4.1}$$

In contrast, beam b) is unable to elongate due to the fixed ends and intuitively the total strain would be zero. From Equation (3.3.2), the mechanical strain ε_σ would therefore be equal and opposite to the thermal strain:

$$-\varepsilon_\theta = \varepsilon_\sigma \tag{3.4.2}$$

This proves that when a restrained beam is heated, the internal mechanical strain is negative, creating a compressive force in the element. So by contrasting beam a) and b) in Figure 3.3, a thermal load will cause strain but no stresses when the member is free from restraint, but will cause stress and no strain when fully restrained. Therefore, the relationship between stress and strain due to thermal loading depends on the restraint.

To simplify the analysis of structures in fire, temperature loads are typically converted into equivalent thermal forces, as at their most fundamental level structural analysis systems consider only force vectors, stiffness matrices and displacement vectors. The thermal stress and forces are assumed to cause the same deflections as the thermal strains would. In other words, a thermal strain is simulated by applying

a equivalent thermal pseudo-stress. To provide framework for understanding these effects the following naming convention has been adopted throughout this research: The symbol N_θ refers to the axial resultant thermal strain load caused by equivalent thermal stresses with the symbol σ_θ, i.e. N_θ simulates the axial elongation of a member due to applied temperatures. These stresses are calculated directly by Equation (3.4.3) from the thermal strains and the Young's Modulus (E_θ), which is adjusted by reduction factors at higher temperature for each element. Codes such as the EN 1993-1-2 (BSI, 2005) define these reduction factors, as explained in Section 2.5.

$$\sigma_\theta = E_\theta \varepsilon_\theta \tag{3.4.3}$$

N_θ is the integration of σ_θ over the area of a section. Applying the fibre section analogy, the calculation can be simplified by taking the sum of the load on each fibre i, as seen in Equation (3.4.4). This is illustrated by an example in Figure 3.4 of a cross-section experiencing uniform heating, where a temperature θ would cause an elongation ε_θ, and this is simulated by applying a thermal stress σ_θ that gives the resultant N_θ.

$$N_\theta = \sum_{i=1}^{n} A_i \sigma_{i,\theta} \tag{3.4.4}$$

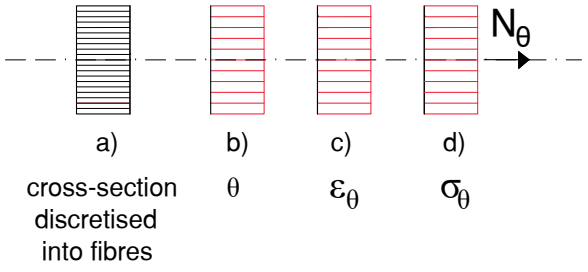


Figure 3.4: Example of a uniformly heated beam, showing a) the rectangular cross-section consisting of multiple fibres, b) the temperature profile θ, c) the thermal strain ε_θ, d) the equivalent thermal stress σ_θ and equivalent thermal force N_θ.

A more complex situation is created when a beam is subjected to bending or non-uniform heating along the height of a cross-section, as illustrated in Figure 3.5. The linear temperature profile θ shown in b), leads to a linear thermal strain ε_θ in c) and equivalent thermal stress σ_θ in d) with an eccentrically applied N_θ. The stress profile can be converted into a N_θ applied at the NA in e) and a resultant thermal strain moment which is given by the symbol M_θ, shown in f). This pseudo-moment leads to curvature of the section and can be determined by the sum of the load on each fibre multiplied by the distance to the NA, as in Equation (3.4.5), where y_i is the distance from the NA to the centroid of fibre i. This is further illustrated in Section 3.5.3.

$$M_\theta = \sum_{i=1}^{n} A_i \sigma_{i,\theta} y_i \tag{3.4.5}$$

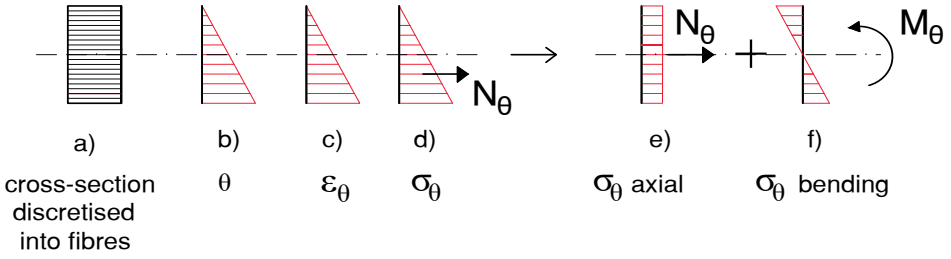


Figure 3.5: Example of a non-uniformly heated beam, showing a) the rectangular cross-section, b) the temperature profile θ, c) the thermal strain ε_θ, d) the equivalent thermal stress σ_θ and equivalent thermal force N_θ at an eccentricity, which is converted into e) N_θ acting at the NA and f) an equivalent thermal moment M_θ.

The thermal loads N_θ and M_θ are not to be confused with the externally applied mechanical forces N and M. Mechanical forces only include the applied mechanical forces due to loads, and induced mechanical forces at the boundary conditions due to restraint.

3.5 Calculation of section stiffness and neutral axis

An important concept in structural fire design is that the stiffness and NA of a section are not necessarily at the same position as at ambient temperature. In this work the NA is defined as the position of zero strain under pure bending. It must be noted that when axial forces are applied the position of zero strain will move, and could even theoretically exist outside of the section. At high temperatures, the material gradually decreases in stiffness. As a result, the NA migrates towards the stiffer part of the section. The calculation of the axial stiffness (EA), bending stiffness (EI) and NA of a section becomes an iterative procedure, as displayed in the flowchart in Figure 3.6 and explained in this section. Note that the original position of the two nodes of the beam element are specified along the reference axis (RA) and the NA is defined at a position relative to this RA. It must be understood that the RA represents the nodes where elements are modelled in a FE program, whereas the NA represents the position about which bending occurs (i.e. point of zero strain when pure bending is applied). The steps below explain a single step of the aforementioned iterative methodology.

3.5.1 Input parameters

Before the iterations for the section stiffness and NA can begin, an initial NA is assumed. Then the sectional properties (A_i) and (I_i), temperature (T), Young's Modulus (E) and yield stress (f_y) of each fibre is specified. The mechanical loading on the section, the applied moment (M) and axial load (N), of the section is required as input. M and N may vary during when global analyses are considered. However, Figure 3.6 represents a cross-sectional analysis that calculates the stiffness of each beam section used in the global analysis.

3.5.2 Iterative steps

The first iteration starts by using an assumed ambient NA and Young's Modulus (E_{20}) and applying the loading around this NA. The strains resulting from the load are calculated. From the strains, the material properties determine the stresses induced. The relationship of strain and stress determines the stiffness and a new secant Young's Modulus ($E_{s,\theta}$) is calculated. As the stiffness is updated, the NA shifts to the stiffer portion of the section and this new position (c'), i.e. the distance from the RA, is determined. Then the next iteration $j+1$ begins and j is incremented until there is no more change in stiffness or shift of NA, which is when convergence is reached.

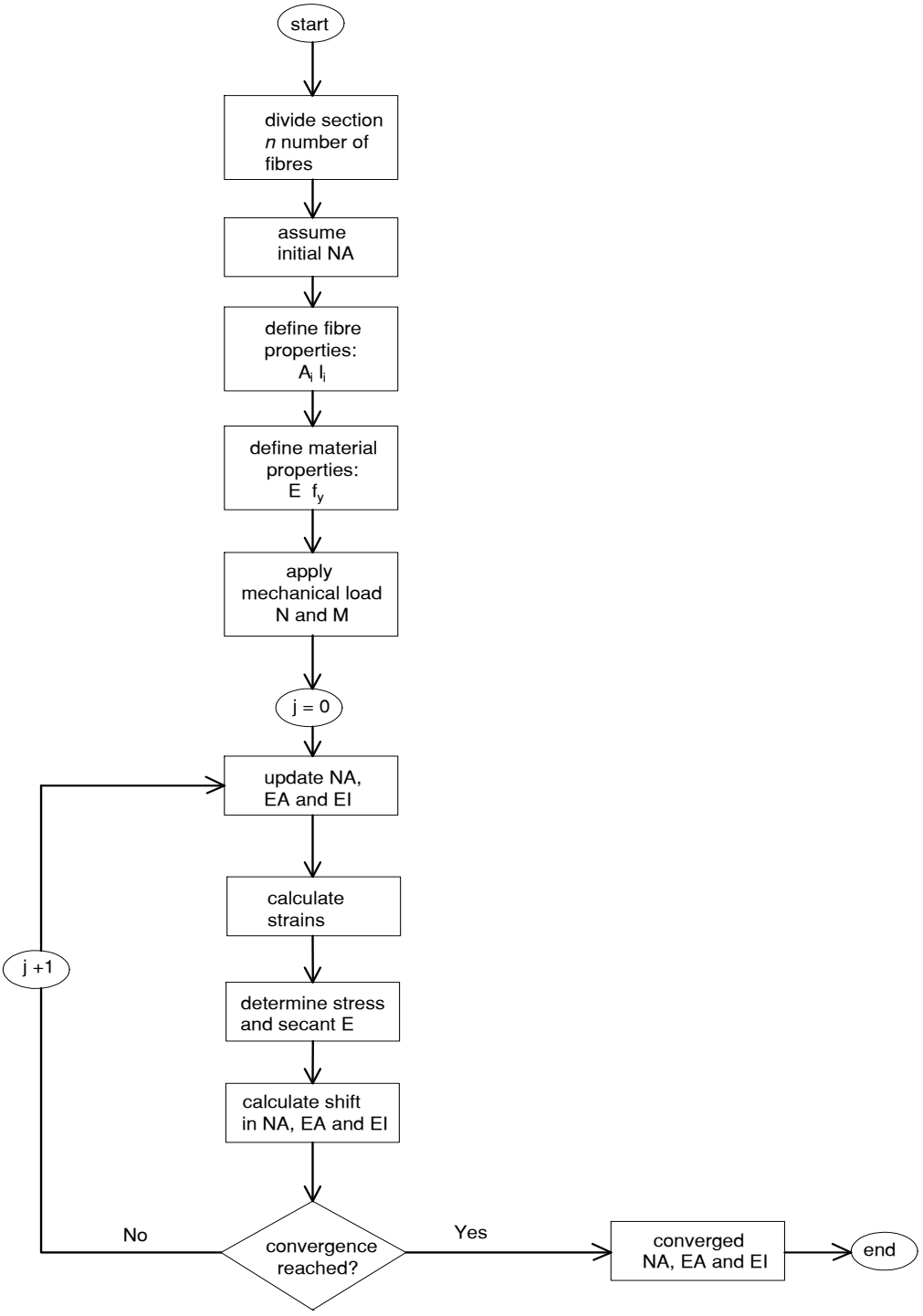


Figure 3.6: Flowchart showing the steps of calculating the section stiffness and NA position

3.5.2.1 Calculation of strains, stresses and Young's Modulus

In each iteration, the strains and stressed need to be recalculated around the updated NA. Firstly, the thermal strain in each fibre is calculated using the specified material model, as explained in Section 3.3. This strain is converted into a equivalent stress and into thermal loads N_θ and M_θ using Equation (3.4.4) and Equation (3.4.5) respectively. Note that the y_i value of each fibre is the distance of the fibre relative to the updated NA. Only once this equivalent thermal loading and the applied mechanical loading, N and M, is defined, can the total strain be determined using the fundamental structural mechanics formula, which is the sum of the axial and bending strain:

$$\begin{aligned} \varepsilon &= \varepsilon_{axial} + \varepsilon_{bending} \\ &= \frac{N + N_\theta}{EA_\theta} + \frac{(M + M_\theta)y}{EI_\theta} \end{aligned} \tag{3.5.1}$$

Rearranging 3.3.2, the mechanical strain is simply the difference between the total and thermal strain, as seen in Equation (3.5.2) and Figure 3.7.

$$\varepsilon_\sigma = \varepsilon - \varepsilon_\theta \tag{3.5.2}$$

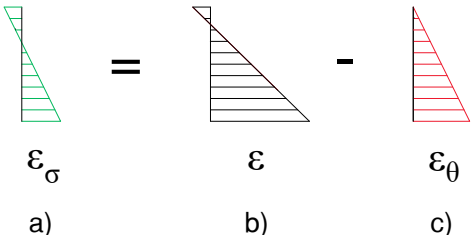


Figure 3.7: Calculation of a) the mechanical strain by the difference of b) the total strains and c) the thermal strain.

The mechanical strain (ε_σ) is important as it is used to determined the value of the mechanical stress (σ_i) and secant elastic modulus ($E_{s,\theta,i}$) for each fibre i. The calculation of σ_i and $E_{s,\theta}$ as functions of (ε_σ) and temperature, and is typically specified in codes such as the Eurocodes (BSI, 2004*a*, 2005), as discussed in Section 2.5.

3.5.2.2 Calculation of section stiffness

Using the updated secant Young's Modulus ($E_{s,\theta}$), the axial and bending stiffness of the section is calculated by the sum of the stiffness of each fibre:

$$E_s A_\theta = \sum_{i=1}^{n} E_{s,i,\theta} A_i \tag{3.5.3}$$

$$E_s I_\theta = \sum_{i=1}^{n} E_{s,i,\theta} I_i \tag{3.5.4}$$

The updated position of the NA (c') relative to the RA for this iteration can be determined by:

$$c' = \frac{\sum_{i=1}^{n} E_{s,i,\theta} A_i y_i}{\sum_{i=1}^{n} E_{s,i,\theta} A_i} \tag{3.5.5}$$

The NA, EA and EI values are updated for the next iteration and the steps are repeated until they do not shift any further. Convergence is reached when the values of c', EA and EI for an iteration are the same as the previous iteration.

3.5.3 Example A

This section provides a simple theoretical structural problem illustrating the basic principles of the shifting NA explained in the previous section. A steel cantilever beam of length (L) 1 m with a rectangular cross section as seen in Figure 3.8, is analysed. The cantilever was chosen as a simple example because it experiences a constant moment throughout the beam and a single beam element is therefore sufficient for analysis. Furthermore, the deflection (δ) of the tip of a cantilever with only a moment applied can be determined analytically by Equation (3.5.6) and even applies to large deflections (Cook *et al.*, 2001). The thermal gradient along the height of a cross-section induces such an equivalent thermal moment and will be modelled as acting at the tip of the beam.

$$\delta = \frac{EI}{M}(1 - cos(\frac{ML}{EI})) \tag{3.5.6}$$

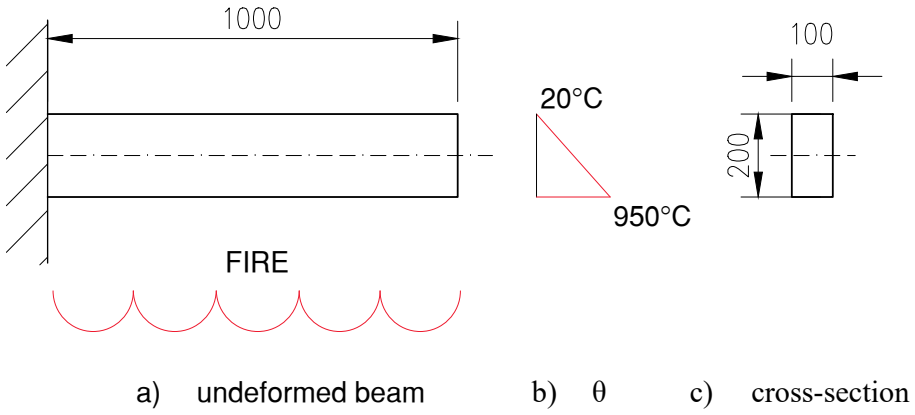


Figure 3.8: Sketch of Example A showing a) the undeformed beam, b) the temperature profile θ and c) the rectangular steel cross-section.

3.5.3.1 Section and material properties

The rectangular steel section has a width of 100 mm and a height of 200 mm. The section is discretised into 20 rectangular elements with a height of 10 mm and a width of 100 mm each. At ambient temperature the steel has a yield stress of 355 MPa and a Young's Modulus of 200 GPa. The reduction factors and material response at elevated temperatures are calculated according to EN 1993-1-2 (BSI, 2005).

3.5.3.2 Thermal loading

In a fire scenario, the bottom of a beam would typically be hotter than the top with a non-uniform temperature distribution (Usmani *et al.*, 2001). However, for simplicity, a theoretical simple linear temperature profile was chosen across the section. The beam was heated from below and a linear thermal gradient of 4.65 °Cmm induced along the height of the cross-section. The upper fibre remains at 20 °C while the bottom fibre is heated to 950 °C, which is approximately the temperature of unprotected steelwork in a 60 min standard fire (refer back to Figure 2.2).

3.5.3.3 Iterations

The NA and stiffness in each case is determined by a number of cross-sectional iterations (as explained in Section 3.5.2). The calculations were done in a Microsoft Excel spreadsheet and applied the formulae as stated in Section 3.5.2. The NA position, thermal load N_θ and M_θ, sectional stiffnesses EI_θ, EA_θ and EAy_θ, and shift relative to the RA (c') for the first iteration (j=0), second iteration (j=1) and last iteration (j=2) are summarised in Table 3.1. Only three iterations were required for convergence of c' and the sectional stiffnesses. The detailed calculation tables in Appendix A have been provided as a simple example against which other developers can compare results.

Table 3.1: Summary of the sectional parameters of Example A calculated for the first ($j = 0$), second ($j = 1$) and last iteration ($j = 2$).

Iteration	$j = 0$	$j = 1$	$j = 2$
NA position (mm)	0	39.3	37.4
N_θ ($\times 10^3$ N)	8408	5540	8408
M_θ ($\times 10^6$ Nmm)	-48.4	114.8	226.4
EI_θ ($\times 10^{12}$ MPa mm^4)	6.93	1.71	3.98
EA_θ ($\times 10^9$ MPa mm^2)	1.441	2.104	2.104
EAy_θ ($\times 10^9$ MPa mm^3)	56.7	78.8	78.8
c' (mm)	39.3	37.4	37.4

3.5.3.4 Strains and stresses

The total strain (ε), thermal strain (ε_θ) and mechanical strain (ε_σ) along the height of the cross-section are represented in Figure 3.9. As the temperature profile is linear, the ε_θ follows an almost linear curve, except at position -50 mm to -75 mm below the RA. This is as expected, since EN 1993-2-1 (BSI, 2005) specifies a constant thermal elongation of 11×10^{-3} in steel for temperatures between 750°C and 860°C (refer to Section 2.5.1.1), which are the temperatures exactly at those points in the cross-section. The ε however, follows a linear curve as the beam will bend according to Bernoulli's principle, resulting in non-linear ε_σ profile accounting for the difference between ε and ε_θ.

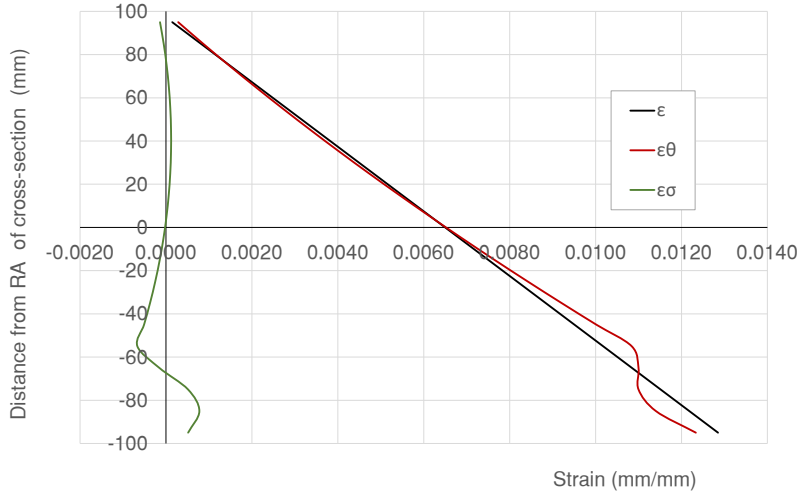


Figure 3.9: Strains along the height of the cross-section of the cantilever beam.

The mechanical stress developed along the height of the cross-section is depicted in Figure 3.10. The mechanical stress σ is calculated from ε_σ and the Young's Modulus (E_θ), which also follows a non-linear profile as a result of temperature and material softening. This example demonstrates that a relatively simple cantilever beam with a rectangular cross-section and linear temperature profile can develop a relatively complex non-linear stress profile. There are alternating bands of tension and compression, making the structural behaviour of the cross-section difficult to analyses using simplified methods.

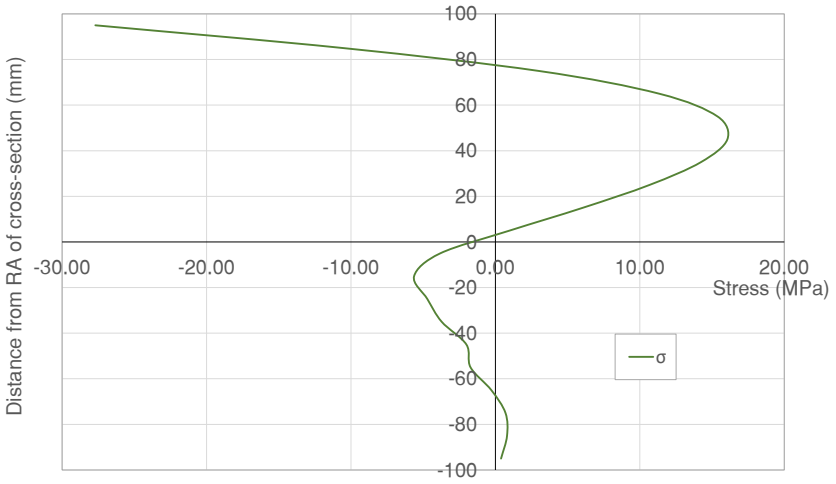


Figure 3.10: Stress developed along the height of the cross-section of the cantilever beam.

3.5.3.5 Deflection of the beam

The effect on the geometry of the thermal gradient can be seen in Figure 3.11. The updated NA position relative to the RA (c') calculated by Equation (3.5.5) converged at a value of 37.4 mm above the centre of the cross-section, as seen in the last iteration ($j = 2$) in Table 3.1. This is as expected since the cooler section is stiffer than the heated section.

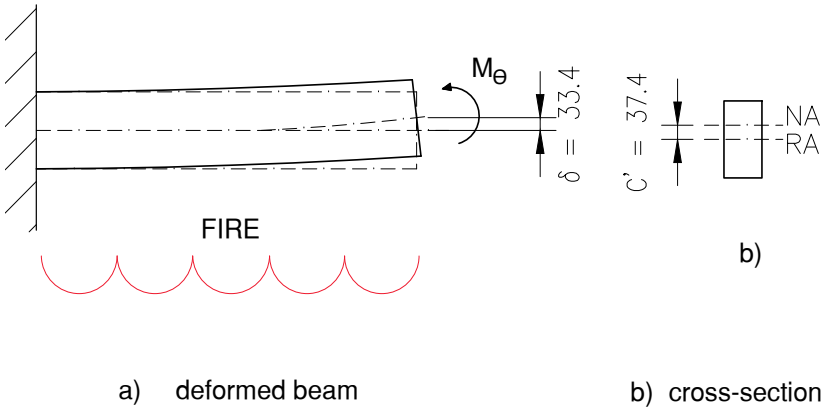


Figure 3.11: Sketch of Example A showing a) the deformed beam and b) the cross-section with the updated NA position relative to the RA.

The thermal gradient also induces a equivalent thermal moment (M_θ), which is calculated as 266.4 kNm using Equation (3.4.5), resulting in compressive strains in the top section and tensile strains in the bottom section. The bending stiffness (EI_θ) around the new NA is calculated as 3.98 MN/m^2. Therefore, the cantilever tip deflects by 33.4 mm, as calculated in Equation (3.5.7) below. Note that the FBE OpenSees model developed in Chapter 4 calculates an identical deflection to this value.

$$\begin{aligned}\delta &= \frac{EI}{M}(1 - cos(\frac{ML}{EI})) \\ &= \frac{3.98 \times 10^{12}}{266.4 \times 10^6}(1 - cos(\frac{266.4 \times 10^6 \times 1000}{3.98 \times 10^{12}})) \\ &= 33.4 \text{ mm}\end{aligned} \tag{3.5.7}$$

3.6 Finite element matrices developed for the FBE methodology

In order to expand the FBE method as a tool for the analysis of statically indeterminate structures in fire, the FBE formulation requires integration into finite element (FE) theory, as the methodology in Section 3.5 above only calculates cross-sectional stiffness about an updated NA position. However, as structures become more complex with restraint and more members, the analysis cannot simply be done by hand-calculation and FEA (finite element analysis) is required.

Any structural static analysis is based upon the well-known relationship between force, stiffness and deformation, as seen in Equation (3.6.1).

$$\{\boldsymbol{F}\} = [\boldsymbol{K}]\{\boldsymbol{\Delta}\} \tag{3.6.1}$$

The global stiffness matrix is represented by $[\boldsymbol{K}]$, the load vector by $\{\boldsymbol{F}\}$ and the displacement vector by $\{\boldsymbol{\Delta}\}$. To incorporate the shift of the NA in this fundamental formulation of FEA, the stiffness matrix and the load vector require modification on a local scale, as explained in the remainder of this section.

3.6.1 Fundamental FE theory for beams

The FBE is based on the classical Euler-Bernoulli beam element with six degrees of freedom (DOFs) as depicted in Figure 3.12 (Cook *et al.*, 2001).

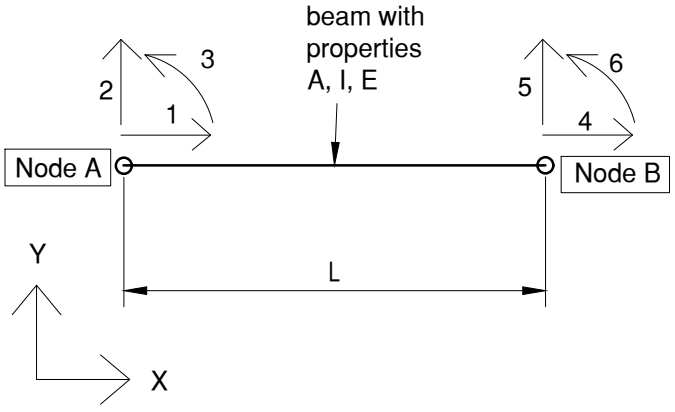


Figure 3.12: Beam element with six DOFs shown

The local load vector $\{\boldsymbol{f_{AB}}\}$ and the displacement vector $\{\boldsymbol{u_{AB}}\}$ of element AB are defined by Equation (3.6.2) and Equation (3.6.3) respectively.

$$\{\boldsymbol{f_{AB}}\} = \begin{Bmatrix} F_{xA} \\ F_{yA} \\ M_{zA} \\ F_{xB} \\ F_{yB} \\ M_{zB} \end{Bmatrix} \tag{3.6.2}$$

$$\{\boldsymbol{u_{AB}}\} = \begin{Bmatrix} u_A \\ v_A \\ \varphi_A \\ u_B \\ v_B \\ \varphi_B \end{Bmatrix} \tag{3.6.3}$$

The local stiffness matrix $[\boldsymbol{k_{AB}}]$ of this beam element is the combination of it's elastic stiffness matrix $[\boldsymbol{k_{el}}]$ and it's geometric stiffness matrix $[\boldsymbol{k_g}]$:

$$[\boldsymbol{k_{AB}}] = [\boldsymbol{k_{el}}] + [\boldsymbol{k_g}] \tag{3.6.4}$$

The properties of the beam that influence the elastic stiffness matrix are the cross-sectional area A, the Young's Modulus E, the second moment of area I of the cross-section and the length L of the beam element, as represented by Equation (3.6.5) (Cook *et al.*, 2001).

$$[\boldsymbol{k_{el}}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ & & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ & & & \frac{EA}{L} & 0 & 0 \\ & sym. & & & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ & & & & & \frac{2EI}{L} \end{bmatrix} \tag{3.6.5}$$

The geometric stiffness matrix plays a role in considering the buckling of the beam element. It is therefore a function of the axial force P and the length L of the beam element (Cook *et al.*, 2001):

$$[\boldsymbol{k_g}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & \frac{6P}{5L} & \frac{P}{10} & 0 & -\frac{6P}{5L} & \frac{P}{10} \\ & & \frac{12PL}{15} & 0 & -\frac{P}{10} & -\frac{PL}{30} \\ & & & 0 & 0 & 0 \\ & sym. & & & \frac{6P}{5L} & -\frac{P}{10} \\ & & & & & \frac{2PL}{15} \end{bmatrix} \tag{3.6.6}$$

3.6.2 Derivation of the modified stiffness matrix and load vector for FBE

In the original FBE work, a matrix manipulation technique to modify the stiffness matrix and load vector of a beam to account for the shift in NA (c') is proposed. The derivation was based on the concept of static condensation, also referred to as the master-slave technique (Cook *et al.*, 2001), and is applied when some of the degrees of freedom (DOFs) of the model (condensed or slave DOFs) are governed by other DOFs specified in the model (retained or master DOFs). This concept can be applied to FBE. The RA is connected by the two master nodes (nodes 1 and 2), which have been specified in the computer model. The updated NA is connected by the two slave nodes (nodes A and B), which are at a distance c' from the master nodes, as depicted in Figure 3.13. It is important to remind the reader that the beam element deforms about the updated NA, but the computer model only recognises the RA. Therefore, the updated NA needs to be slaved to the RA.

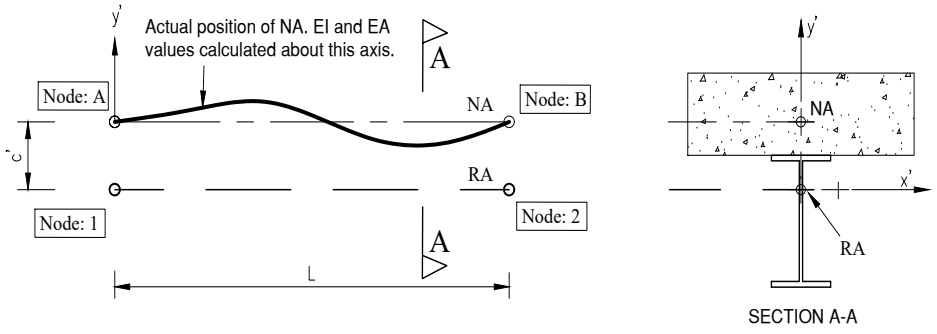


Figure 3.13: Beam element with reference axis (RA) and updated axis (NA) shown.

At a certain time t, the beam is already deformed and a local axis system is applied. The FBE therefore follows the corotational formulation (Iu *et al.*, 2005) to consider non-linear behaviour. This means that the geometry is updated at each iteration, to account for the deformation that already occurred in the previous steps. The deflection $\{\boldsymbol{\Delta u}\}$ of the nodes 1 and A at a time t to time $t + \Delta t$ is shown in Figure 3.14 and their relationship defined by Equation (3.6.7).

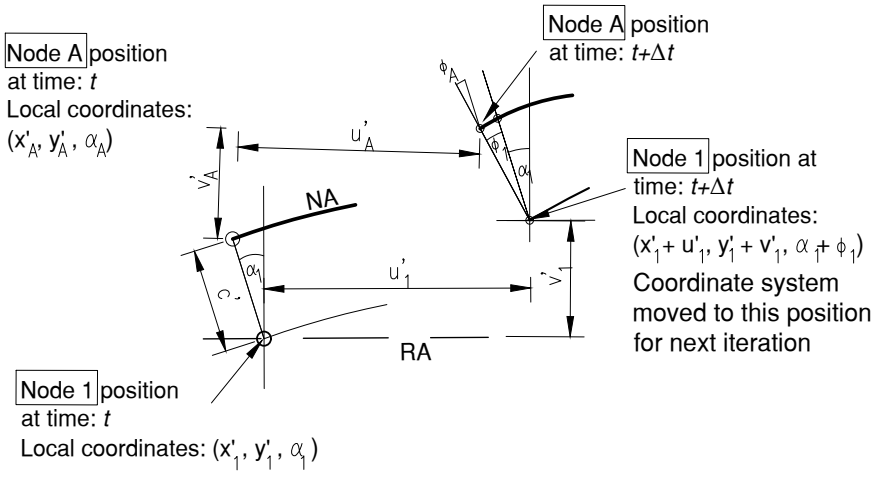


Figure 3.14: Geometry of RA and the deformed configuration, showing the positions of nodes 1 and A at time t and $t + \Delta t$.

$$\{\boldsymbol{\Delta u_A}\} = \begin{Bmatrix} \Delta u_A \\ \Delta v_A \\ \Delta \varphi_A \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \Delta u_1 \\ \Delta v_1 \\ \Delta \varphi_1 \end{Bmatrix} + c' \begin{Bmatrix} -sin(\alpha_1 + \varphi_1) + sin(\alpha) \\ cos(\alpha_1 + \varphi_1) - cos(\alpha_1) \\ 0 \end{Bmatrix}$$
$$\approx \begin{bmatrix} 1 & 0 & -c' \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \Delta u_1 \\ \Delta v_1 \\ \Delta \varphi_1 \end{Bmatrix} = [\boldsymbol{Q}]\{\boldsymbol{\Delta u_1}\} \tag{3.6.7}$$

The coordinate vectors of Node A $\{\boldsymbol{x_A}\}$ and Node 1 $\{\boldsymbol{x_1}\}$ are linked according to the following relationship:

$$\{\boldsymbol{x_A}\} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ \alpha_1 \end{Bmatrix} + c' \begin{Bmatrix} -sin(\alpha_1) \\ cos(\alpha_1) \\ 0 \end{Bmatrix} \approx \begin{bmatrix} 1 & 0 & -c' \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ \alpha_1 \end{Bmatrix} + \begin{Bmatrix} 0 \\ c' \\ 0 \end{Bmatrix}$$
$$= [\boldsymbol{Q}]\{\boldsymbol{x_1}\} + \begin{Bmatrix} 0 \\ c' \\ 0 \end{Bmatrix} \tag{3.6.8}$$

Similarly, the local forces at Node A and Node 1 can also be linked:

$$\{\boldsymbol{f_A}\} = \begin{Bmatrix} F_{xA} \\ F_{yA} \\ M_{zA} \end{Bmatrix} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ M_{z1} + c' F_{x1} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c' & 0 & 1 \end{bmatrix} \begin{Bmatrix} F_{x1} \\ F_{y1} \\ M_{z1} \end{Bmatrix} = [\boldsymbol{W}]\{\boldsymbol{f_1}\} \tag{3.6.9}$$

Note than small deflection and small rotation assumptions are applied, where $sinx \approx x$, which allows for the simplification in Equation (3.6.7) to Equation (3.6.9). The

3x3 matrices $[\boldsymbol{Q}]$ and $[\boldsymbol{W}]$ are operators, with entries c', 1 and 0, that link the displacement and force of a node on the updated NA to the RA. These matrices can be expanded to 6x6 matrices $[\boldsymbol{Q_{AB}}]$ in Equation (3.6.10) and $[\boldsymbol{W_{AB}}]$ in Equation (3.6.11) operating on an element linking nodes A and B.

$$[\boldsymbol{Q_{AB}}] = \begin{bmatrix} [\boldsymbol{Q}] & [\boldsymbol{0}] \\ [\boldsymbol{0}] & [\boldsymbol{Q}] \end{bmatrix} \tag{3.6.10}$$

$$[\boldsymbol{W_{AB}}] = \begin{bmatrix} [\boldsymbol{W}] & [\boldsymbol{0}] \\ [\boldsymbol{0}] & [\boldsymbol{W}] \end{bmatrix} \tag{3.6.11}$$

Mathematically, the relationship between the operators is such that the transpose of $[\boldsymbol{Q_{AB}}]$ is equal to the inverse of $[\boldsymbol{W_{AB}}]$:

$$[\boldsymbol{Q_{AB}}]^{\boldsymbol{T}} = [\boldsymbol{W_{AB}}]^{-\boldsymbol{1}} \tag{3.6.12}$$

Consider the element AB with the fundamental local force-displacement relationship as follows:

$$\{\boldsymbol{f_{AB}}\} = \begin{Bmatrix} \boldsymbol{f_A} \\ \boldsymbol{f_B} \end{Bmatrix} = [\boldsymbol{k_{AB}}] \begin{Bmatrix} \boldsymbol{u_A} \\ \boldsymbol{u_B} \end{Bmatrix} \tag{3.6.13}$$

Substituting Equation (3.6.7) and Equation (3.6.9) into Equation (3.6.13):

$$\{\boldsymbol{f_{AB}}\} = [\boldsymbol{W_{AB}}] \begin{Bmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \end{Bmatrix} = [\boldsymbol{k_{AB}}][\boldsymbol{Q_{AB}}] \begin{Bmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{Bmatrix} \tag{3.6.14}$$

Multiplying the above by the inverse of $[\boldsymbol{W_{AB}}]$ gives:

$$\{\boldsymbol{f_{12}}\} = [\boldsymbol{W_{AB}}]^{-\boldsymbol{1}}[\boldsymbol{k_{AB}}][\boldsymbol{Q_{AB}}] \begin{Bmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{Bmatrix} \tag{3.6.15}$$

Applying Equation (3.6.12) to Equation (3.6.15) yields:

$$\{\boldsymbol{f_{12}}\} = [\boldsymbol{Q_{AB}}]^{\boldsymbol{T}}[\boldsymbol{k_{AB}}][\boldsymbol{Q_{AB}}] \begin{Bmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{Bmatrix} \tag{3.6.16}$$

Therefore, the stiffness matrix of the element around the reference axis $[\boldsymbol{k_{12}}]$ is formulated by:

$$[\boldsymbol{k_{12}}] = [\boldsymbol{Q_{AB}}]^{\boldsymbol{T}}[\boldsymbol{k_{AB}}][\boldsymbol{Q_{AB}}] \tag{3.6.17}$$

Similarly, from Equation (3.6.9) the load vector around the reference axis $\{\boldsymbol{f_{12}}\}$ is also updated by:

$$\{\boldsymbol{f_{12}}\} = [\boldsymbol{W_{AB}}]^{-\boldsymbol{1}}\{\boldsymbol{f_{AB}}\} = [\boldsymbol{Q_{AB}}]^{\boldsymbol{T}}\{\boldsymbol{f_{AB}}\} \tag{3.6.18}$$

The relationships shown in Equation (3.6.17) and Equation (3.6.18) are important results when implementing the FBE formulation in FEA. For each iteration in a FEA, the stiffness and load of each element are updated by calculating the stiffness and load about the updated NA and then manipulating it back to the RA. This means that stiffnesses are not incorrectly calculated about the wrong NA position (i.e. the RA), but the model can still have a fixed RA position consistent with how analyses are typically carried out.

3.6.3 Non-linear analysis and unbalanced forces

As explained in Chapter 2, the analysis of structures in fire is a non-linear problem and a iterative technique is required. Typically, Newton-Raphson or the Modified Newton methods are applied for fire analysis. These methods apply the incremental-iterative method with load control. The load is divided into increments and applied in a number of steps. For each step, the unbalanced forces are calculated and iterations performed until convergence is reached when equilibrium is satisfied. For more information regarding these non-linear solvers refer to McGuire *et al.* (2000) and Bhatti (2016).

3.6.4 Assembling and solving the global structural equation

After the local stiffness matrices and load vectors of each element i in the structure have been modified as explained in the previous section, the method continues with classical FEA methodologies of assembling the global stiffness matrix $[\boldsymbol{K}]$ and the global force vector $\{\boldsymbol{F}\}$. $[\boldsymbol{K}]$ is the conglomeration of the constituent modified stiffness matrices $[\boldsymbol{k_{12,i}}]$ of each element i in the structure (Cook *et al.*, 2001) and is formulated by Equation (3.6.19), where $[\boldsymbol{T_i}]$ represents the transformation matrix from the local to the global coordinate system of each element.

$$[\boldsymbol{K}] = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} [\boldsymbol{T_i}]^{\boldsymbol{T}} [\boldsymbol{k_{12,i}}][\boldsymbol{T_i}] \tag{3.6.19}$$

Similarly, $\{\boldsymbol{F}\}$ is the assembly of the constituent modified load vectors $\{\boldsymbol{f_{12,i}}\}$ of each element i:

$$\{\boldsymbol{F}\} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} [\boldsymbol{T_i}]^{\boldsymbol{T}} \{\boldsymbol{f_{12,i}}\} \tag{3.6.20}$$

The analysis continues as a typical FEA analysis, by applying the boundary conditions solving the deflections from Equation (3.6.1).

3.7 Application of the FBE methodology to global structural analyses

The theory derived in Section 3.6 can be further applied to entire structures. The FBE methodology aims to simplify a complicated structure, such as seen in Figure 3.15 a) into a skeletal beam model as depicted by Figure 3.15 d). This analysis philosophy is discussed below.

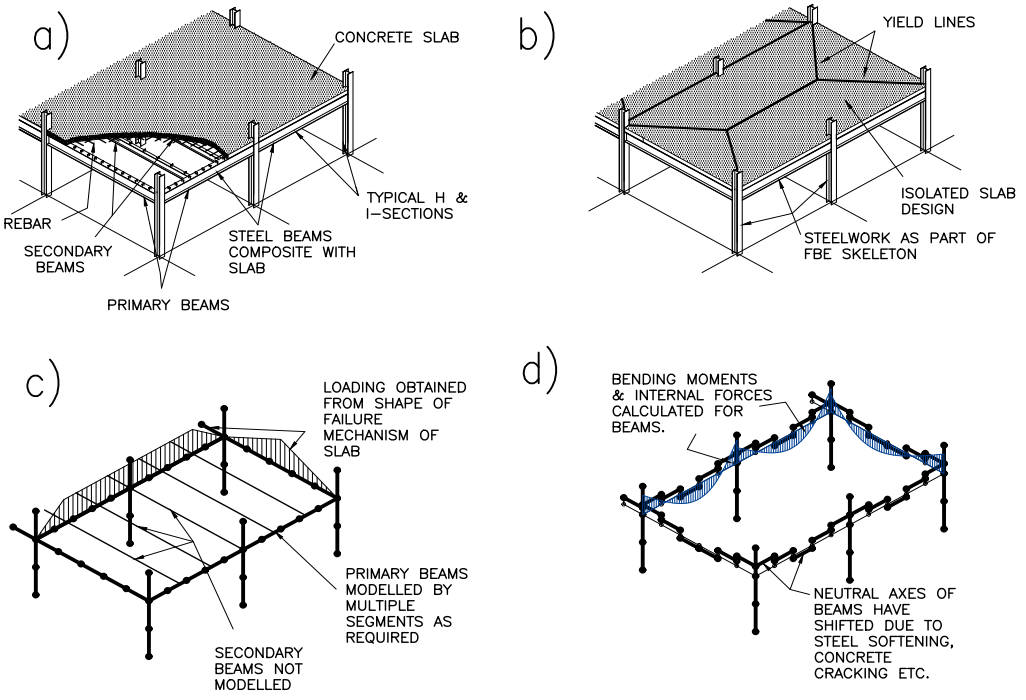


Figure 3.15: Summary of the FBE design steps being applied to a composite structure simplified into an equivalent skeletal frame, reproduced from Walls *et al.* (2018).

Figure 3.15 a) depicts a typical composite structure with steel columns, primary and secondary steel beams and a concrete slab. Diagram b) shows the yield lines of the floor, which is modelled in isolation by methods such as the tensile membrane method by Bailey (2004) or the Slab Panel Method (SPM) by Clifton (2006). The loading of the slabs on the beams is extracted from the SPM results and used as input for the skeletal FBE model, as shown in Figure 3.15 c), which only consists of the primary beams and columns. The secondary beams are considered as part of the slab. The loading on the skeleton would typically include the mechanical loads from the slab and thermal loading, which is applied in the form of temperatures specified. The load is applied in increments and each load cycle is an iterative procedure of updating geometry, strains, stresses and the NA, as discussed in Section 3.5. Once convergence is reached, the bending moments and internal forces can be determined for the beams, as shown in Figure 3.15 d). These forces can then be used for design.

3.8 Advantages and limitations of the FBE formulation

The FBE formulation is a compromise between simple and quick fire analysis methods and detailed, sophisticated and complex analyses. It method aims to be computationally efficient and intuitive such as the simple methods, while still employing fundamental structural mechanics, and being sufficiently accurate such as the complex analyses. The FBE can predict most global structural behaviour in a fire, but it has also limitations due to the nature of a simplified approach.

3.8.1 Benefits of FBE method

The main advantages that the FBE has in relation to prescriptive methods and simple single-element analyses are that the results are generally more accurate and true to structural behaviour in fire. These advantages are listed below:

- Analyses are not limited to individual elements but can be applied to skeletal frames.
- Temperature gradients along the height of a section can be modelled using fibres.
- Localised fires and travelling fires can be modelled.
- The FBE considers the shift in NA, which is not done in a simple Euler-Beam element.
- Failure can potentially be predicted from the stresses in the members or using design codes with the calculated internal forces.
- The FBE method can predict run-away failure by showing non-convergence, although further research is required in this area.
- The FBE cross-sections include non-linear material behaviour and cracking.
- The method is not limited to concrete or steel, but can be applied to any other construction material.

On the other end of the spectrum of structural fire analyses lie the advanced modelling techniques, which typically use FEA software such as SAFIR, Vulcan or ABAQUS (refer to Section 2.7). Compared to these complex methods, the FBE methods has the following advantages:

- The FBE method demands less time of the modeller as the structure is simplified into a skeletal frame.
- The computation time of the analysis is greatly reduced. This is mainly because of the reduced number of global DOFs. For example, beam elements have less DOFs than shell elements, which are typically used in advanced analyses.
- The formulation is derived from first principles and is versatile in it's application.

- The FBE formulation can be integrated in a FEA analysis by the modification of stiffness matrices and load vectors, as presented in Section 3.6.2.
- Composite structures can be modelled using single beam elements for a cross-section without the use of shell elements, which greatly increase the DOF of a model.
- Case studies performed, as discussed further in Chapter 5, have shown that the FBE formulation predicts results comparable to advanced FEA computer models.

3.8.2 Limitations of the FBE method

As in any analysis tool, there are some structural phenomena which the FBE formulation does not consider or cannot capture:

- The FBE, in its current form, does not take into account slippage between concrete and steel elements.
- Shear deformations and localised distortions are neglected.
- The method becomes less accurate when Euler-Bernoulli assumptions are violated i.e. plane sections need to remain plane.
- The analysis is only applicable for pre-buckling behaviour. The FBE method cannot predict local buckling. It would need to be coupled with existing design codes for the consideration of buckling behaviour.
- Floor slabs have to be modelled separately by a different method. However, Walls (2016) suggested that the FBE method would be powerful in conjunction with the SPM developed by Clifton (2006). The skeleton frame would be modelled using FBE and the floors using SPM. For more detail regarding the SPM method refer to Clifton (2006); Clifton and Abu (2014) and Walls *et al.* (2017).
- Similarly, catenary action of secondary beams is not explicitly considered, but can be incorporated when applied with the SPM.
- The FBE is more complicated than the Euler-Bernoulli beam as additional iterations regarding the shift of the NA is required. However, the computation is still quicker than models with shells or volume elements.
- The width of the concrete flange in a composite element is not calculated by FBE. This needs to be pre-determined.

3.9 Conclusion

This chapter has provided an overview of the basic FBE theory which is applicable to this research, and is based upon the previous work in which the FBE was developed. Firstly, the effects of thermal strains developed in structures subjected to fire was discussed and shown how these are converted to equivalent thermal forces. Furthermore, the shifting of the NA and the change in stiffness due to a member exposed to fire has been explained and portrayed by a simple cantilever example. The integration of the FBE into FEA has been derived and shown that by manipulating the stiffness matrices and load vectors, the FBE formulation can be elegantly implemented into FEA. The design steps of a 3D analysis of a composite structure were illustrated to show the role of the FBE method in a global analysis. Lastly, the advantages and disadvantages of the FBE methodology were outlined. Building on the theory presented in this chapter, the practical implementation of this method in FEA software is explained in the following chapter.

Chapter 4

The implementation of the Fire Beam Element methodology into the OpenSees finite element software

4.1 Introduction

This chapter discusses how the theory of the previous chapter is applied in Finite Element Analysis (FEA) software to make the Fire Beam Element (FBE) methodology readily available as a tool for modelling global 2D structures in fire. The FBE method was developed in OpenSees software by creating new classes and modifying existing classes in OpenSees for Fire.

This chapter introduces OpenSees by explaining how a model is built for analysis and providing the reasons for choosing the software for this research. The next section explains how the FBE methodology was converted into code written for the framework of OpenSees. Lastly, the FBE application for users of OpenSees is briefly outlined.

The extension of the FBE methodology in OpenSees addresses a number of the shortcomings in the previous version of the work, in that now global analyses of structures can be carried out, as will be illustrated in Chapter 5. It is important to note that within OpenSees, the framework already has functions to consider sections with multiple fibres making up a cross-section. However, the important contribution of this work is the inclusion of the functions to account for a moving, eccentric neutral axis (NA), as outlined in the previous chapter. As an additional contribution, the number of temperature points along the height of a cross-section have been increased.

4.2 OpenSees software

4.2.1 Introduction to OpenSees

OpenSees is an open-source software supported by a large community of users and developers who collaborate as a combined effort towards research and development in structural engineering. The software was developed at the University of California, Berkeley, in conjunction with PEER and Nees, and the original use was for earthquake analysis (McKenna, 1997). However, since then OpenSees has become a powerful tool for the analysis of any non-linear response for structures.

In 2009 the University of Edinburgh started the development of "OpenSees for Fire" as a long-term project of students and researchers. A significant part of their work has been focussed on heat transfer and thermo-mechanical analyses (Jiang *et al.*, 2015). This included the implementation of material models at elevated temperatures from the Eurocodes (BSI, 2004*b*, 2005) as well as thermal load classes. For more details regarding OpenSees for Fire refer to Jiang and Usmani (2013); Jiang *et al.* (2014, 2015); Jiang and Usmani (2018).

4.2.2 Analysis model in OpenSees

As in any process, the analysis of a structure in OpenSees can be divided into the input, analysis and output components, as summarised in Figure 4.1. The input for OpenSees is provided in the form of a Tool Command Language (TCL) script. The analysis is performed by running code written in object-orientated language C++. In this research, Miscrosoft Visual Studio 2017 was used as a platform to run the code. Finally, the output of the analysis is given in the form of output files, either as a Comma-Separated Values (CSV), Extensible Markup Language (XML), binary or data file. This work adopted the CSV files for output.

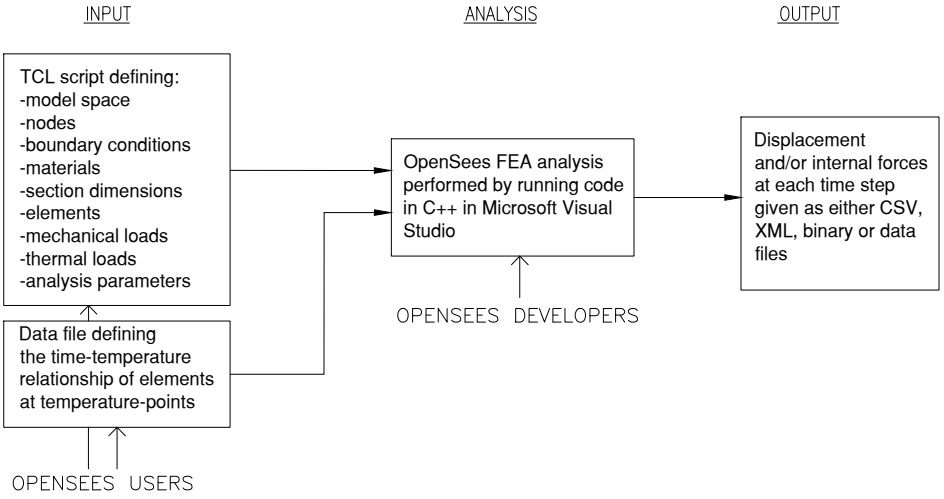


Figure 4.1: Flowchart showing the input, analysis and output components of a FEA analysis in OpenSees.

4.2.2.1 TCL input for OpenSees

A user of OpenSees would typically only be required to write an input script in TCL that gets called by OpenSees. Various user-friendly commands are used to build the model, which includes defining the model space, the positions of the nodes, the materials, the cross-sections, the boundary conditions, the element types, mechanical and thermal loads as well as analysis parameters such as the convergence tolerance or type of non-linear solver used. Most material models from the Eurocodes are built-in functions.

An example of such a TCL input script can be found in Appendix B.1. This script is the input for an analysis of a simply-supported beam in a furnace with non-uniform heating along its length as well as a horizontal and vertical point load. This example was used as a case study and will be analysed in detail in Section 5.2. For the command manual for writing TCL scripts in OpenSees refer to Jiang (2017).

The time-temperature relationship of a beam element is sent to OpenSees using a data file containing the pseudo-time followed by the temperature at a specified point along the height of a cross-section. This allows for the simulation of a fire by applying a time-temperature history to the structure.

4.2.2.2 C++ framework in OpenSees

The FEA analysis accepts the input file and analyses it based on code written in C++, a powerful and efficient object-orientated programming language. OpenSees includes set frameworks in C++ to use for developers (Jiang and Usmani, 2013), encouraging collaboration in developing the source code of OpenSees. In other words, developers can create their own materials, sections, elements, load types, analysis methods etc by writing code in C++ in the OpenSees platform. An example of the Microsoft Visual Studio platform for developing OpenSees is seen in Appendix B.2.

4.2.3 Reasons for using OpenSees in this research

OpenSees was chosen as the software for the implementation of the FBE concept for various reasons:

- OpenSees is a powerful FEA and non-linear analysis tool.
- OpenSees is open-source and researched-based, which makes it an applicable platform for developing the FBE.
- The software has been extensively tested, for example by Jiang and Usmani (2013); Jiang *et al.* (2014)
- The source code can be accessed and modified in Microsoft Visual Studio. This means that new classes and functions can be developed. It also allowed for the modification of stiffness matrices and load vectors. This is usually impossible or difficult in typical commercial software.

- Temperature-dependent materials, thermal-load classes and thermal-mechanical analysis procedures have already been developed and built into OpenSees for Fire.
- Important concepts of the FBE such as the use of fibre sections, the corotational approach and the Newton-Raphson method are given as options in OpenSees.

4.3 Implementation of the FBE methodology in OpenSees

The focus of this research was the implementation of an intelligent beam element as an object in OpenSees, with functions based on the FBE theory as discussed in Chapter 3. This beam element class was called *FireEl* and linked to a cross-section class *FireFiberSection*, and then integrated into the thermal-mechanical analysis in OpenSees. Both classes were written by the author in C++ as part of the OpenSees platform, which is developed within Microsoft Visual Studio.

4.3.1 Subclasses developed in OpenSees for Fire

The OpenSees C++ framework makes use of hierarchical inheritance, where more than one subclass can be derived from a single base class. A base class, for example *Material*, is defined as the object in the code which acts as the skeleton for all classes of that type (all materials), defining essential variables (for example, the Young's Modulus) and certain functions (for example, the calculation of thermal elongation in the material). The subclasses derived from this base class contain the same variables and functions of the base class but define them in their unique way. For example, the material *SteelECThermal* is a subclass of the base class *Material* and defines the elevated temperature Young's Modulus and thermal elongation according to EN 1993-1-2 (BSI, 2005).

In the same manner, the OpenSees for Fire project developed subclasses in OpenSees to account for parameters specific to structures in fire. This includes classes for applying thermal loads and material classes with temperature-dependent properties, defined according to EN 1993-1-2 (BSI, 2005) for steel and EN 1992-1-2 (BSI, 2004*b*) for concrete. The subclasses for the FBE formulation developed in this research (marked in red) are based on these existing thermal classes and link to the OpenSees inheritance hierarchy, as seen in Figure 4.2.

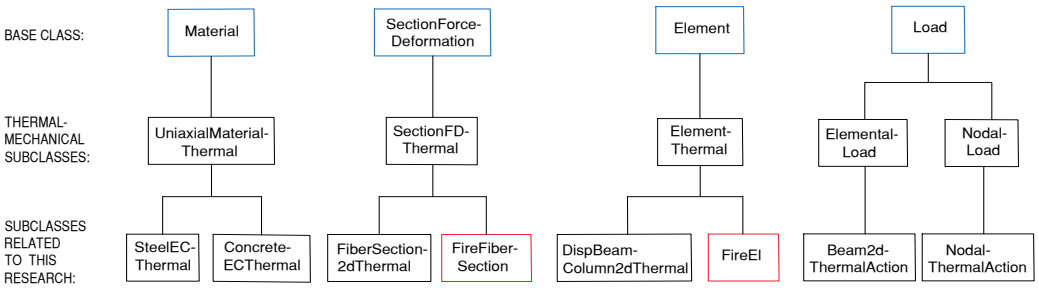


Figure 4.2: Inheritance hierarchy in OpenSees for Fire. The classes in blue are the base classes, while the blocks in black are the thermal-mechanical classes developed by Jiang and Usmani (2013). The blocks marked in red are the subclasses for the FBE formulation developed in this research.

Currently, OpenSees performs a decoupled analysis, which means that the thermal analysis and the structural analysis are done separately. This requires temperature as an input for the mechanical analysis. OpenSees for Fire has therefore developed thermal load classes to store a maximum of 9 temperature points along the height of a cross-section. From the distribution, interpolation is used to assign a temperature to each fibre specified in the section (Jiang and Usmani, 2013).

To further explain the work done in this research to the reader, the functions developed are represented by flowcharts in the following sections. In these flowcharts, the steps in black are performed in the class function itself, while the green represents the classes that are called within the functions to provide input or a calculation to assist the method. The steps marked in red are those that are specifically novel to the FBE methodology.

4.3.2 Development of the class *FireFiberSection*

A fundamental concept of the FBE formulation is the shift in NA (c') relative to the reference axis (RA), which requires updating every iteration. In order to to calculate this shift, the fibre section analogy (as discussed in Section 3.2) was used by the implementation of a object called *FireFiberSection*. This is a subclass of the base class *SectionForceDeformation*, which models the cross-section of an element. *FireFiberSection* performs similar functions to the existing class *FiberSection2dThermal* developed by Jiang and Usmani (2013). Note that the current *FiberSection2dThermal* in OpenSees updates the NA by calculating the geometric centroid of a section by:

$$ybar = \frac{\sum_{i=1}^{n} A_i y_i}{\sum_{i=1}^{n} A_i} \tag{4.3.1}$$

This shift, however, neglects the difference in stiffness of the materials, especially in composite sections and when temperature gradients are applied to the cross-section. Therefore, *FireFiberSection* calculate the shift of the NA by considering

the change in stiffness of a cross-section by the function *calculateC*. Another difference is that *FireFiberSection* is programmed to receive more temperature points along the height of a cross-section.

4.3.2.1 Increasing the number of temperature points

In OpenSees for Fire, a fire load is simulated by assigning input temperatures to a section as the load type *BeamThermalAction*. The original OpenSees for Fire programme accepts 2, 5 or 9 temperature data points which can be specified along the height of a beam's section (Jiang, 2017) and then uses linear interpolation for the calculation of fibre temperatures between the points. However, this was found to be insufficient for the purpose of the FBE formulation, especially when considering composite sections. The temperature distribution along the height of the concrete slab could show great non-linearity, as shown in Figure 4.3 b), and representing this by only 9 points (as is Figure 4.3 c)) would reduce accuracy. Therefore, the number of input temperature points was changed to 25 points along the height of the section, as shown in Figure 4.3 d). The fibre temperature will be determined by linear interpolation between the temperature points. The code for this can be seen in the function *determineFiberTemperature* in the class *FireFiberSection* presented in Appendix C.1.1.

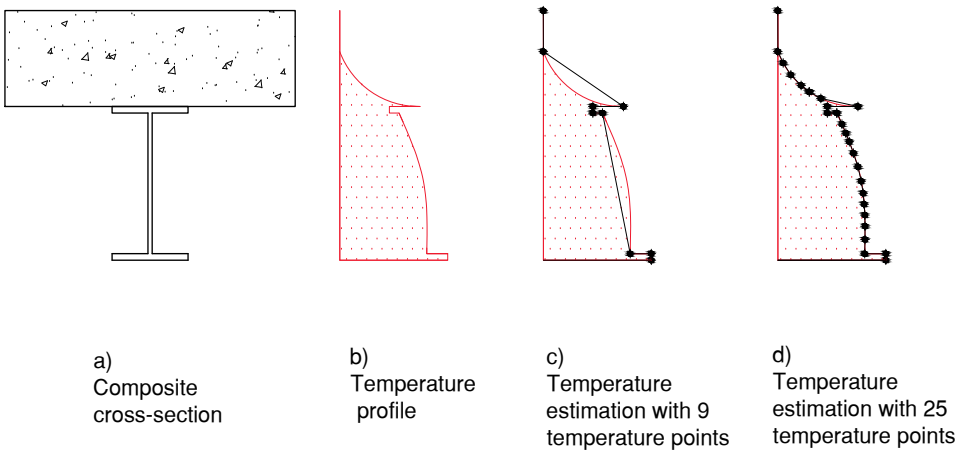


Figure 4.3: A composite cross-section a) with b) a typical non-linear temperature distribution in red, c) showing the 9 point temperature estimation and d) showing the 25 point temperature estimation

4.3.2.2 Calculation of the updated NA position

The shift in NA is determined by the function *calculateC* and based on the theory discussed in Section 3.5. This function is called by the element when the position of the NA relative to the reference axis (RA, as explained in Section 3.5) (c') requires updating in each iteration of an analysis performed. The function can be represented by the flowchart shown in Figure 4.4 and the code in given in Appendix C.1.2. Firstly, the original geometric centroid of the section relative to the RA (*ybar*) is used as

input and the sectional values EA and EAy are initialised. Then the function iterates through n number of fibres, where the temperature T_i, the position y_i relative to the RA, mechanical strain $\varepsilon_{\sigma,i}$ and stress σ_i of each fibre i is determined. The sum of all fibre stiffnesses EA_i and EAy_i are added to the sectional EA and EAy. The value of c' can then be calculated by Equation (4.3.2) as explained in Section 3.5, where E_s is the secant Young's Modulus and A_i is the area of each fibre.

$$c' = \frac{\sum_{i=1}^{n} E_{s,i,\theta} A_i y_i}{\sum_{i=1}^{n} E_{s,i,\theta} A_i} \tag{4.3.2}$$

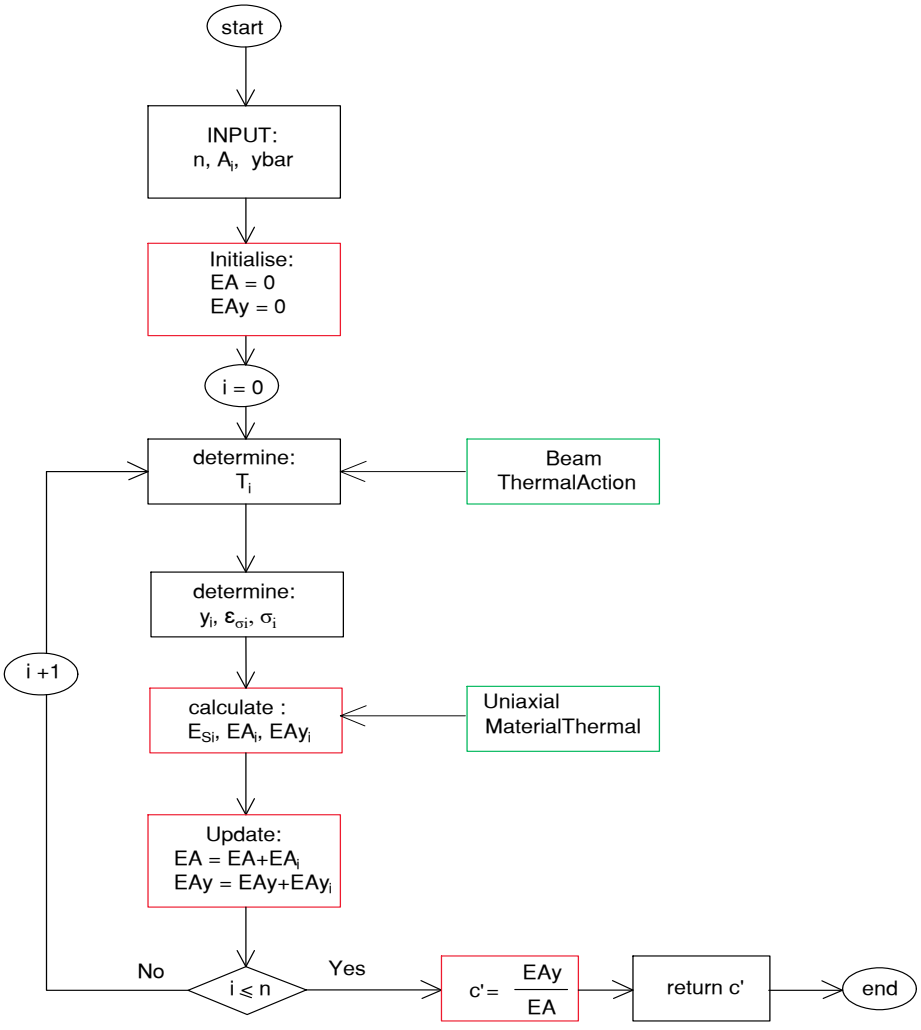


Figure 4.4: Steps of the function *calculateC* in the class *FireFiberSection*

4.3.3 Development of the class *FireEl*

The *FireEl* class written by the author is a subclass of the base class *Element* in OpenSees. In the constructor of the *FireEl* object, the element is linked to two *Node* objects (referred to as Node 1 and 2), a *FireFiberSection* object and a *CorotCrdTransf* object which accounts for corotational geometric transformation (discussed in Section 4.3.3.1). Note that the subscript 12 refers to the nodes on the RA which are defined in the computer model, while the subscript AB refer to the nodes on the updated NA, as explained previously in Section 3.6.2. Important functions that encompass the FBE methodology and therefore discussed in this section are *getTangentStiff* and *getLoad*.

4.3.3.1 Geometric transformation classes in OpenSees

Geometric-transformation objects in OpenSees are responsible for the transformation of beam element stiffness and resisting forces from the local to the global coordinate system. Three different types of transformation options that are currently available in OpenSees are *Linear*, *PDelta* or *Corotational*. As discussed previously in Section 3.6.2, the corotational approach is preferred for the FBE formulation. The existing class *CorotCrdTransf* can therefore easily be linked to the *FireEl* class in OpenSees.

4.3.3.2 Stiffness of the *FireEl*

The updating of the stiffness matrix by considering the updated position of the NA (c') in each iteration is fundamental in the implementation of the FBE methodology. In OpenSees, the assembling and the updating of the element stiffness matrix is done in the function *getTangentStiff*. The value of c' has been incorporated by matrix manipulation as derived in Section 3.6.2. The steps of the function *getTangentStiff* are shown in Figure 4.5 and explained in this section. The actual C++ code is given in Appendix C.2.1.

The function starts by calling *FireFiberSection* to calculate and update the c' value. As an input, the deformed length L is calculated by the corotational coordinate transform class. The calculation of the global stiffness matrix $[\boldsymbol{K}]$ can then be subdivided into the calculation of the local elastic stiffness matrix $[\boldsymbol{k_{el}}]$ and the local geometric stiffness matrix $[\boldsymbol{k_g}]$, as explained in Section 3.6. These are modified separately, as seen by the two branches of the flowchart in Figure 4.5.

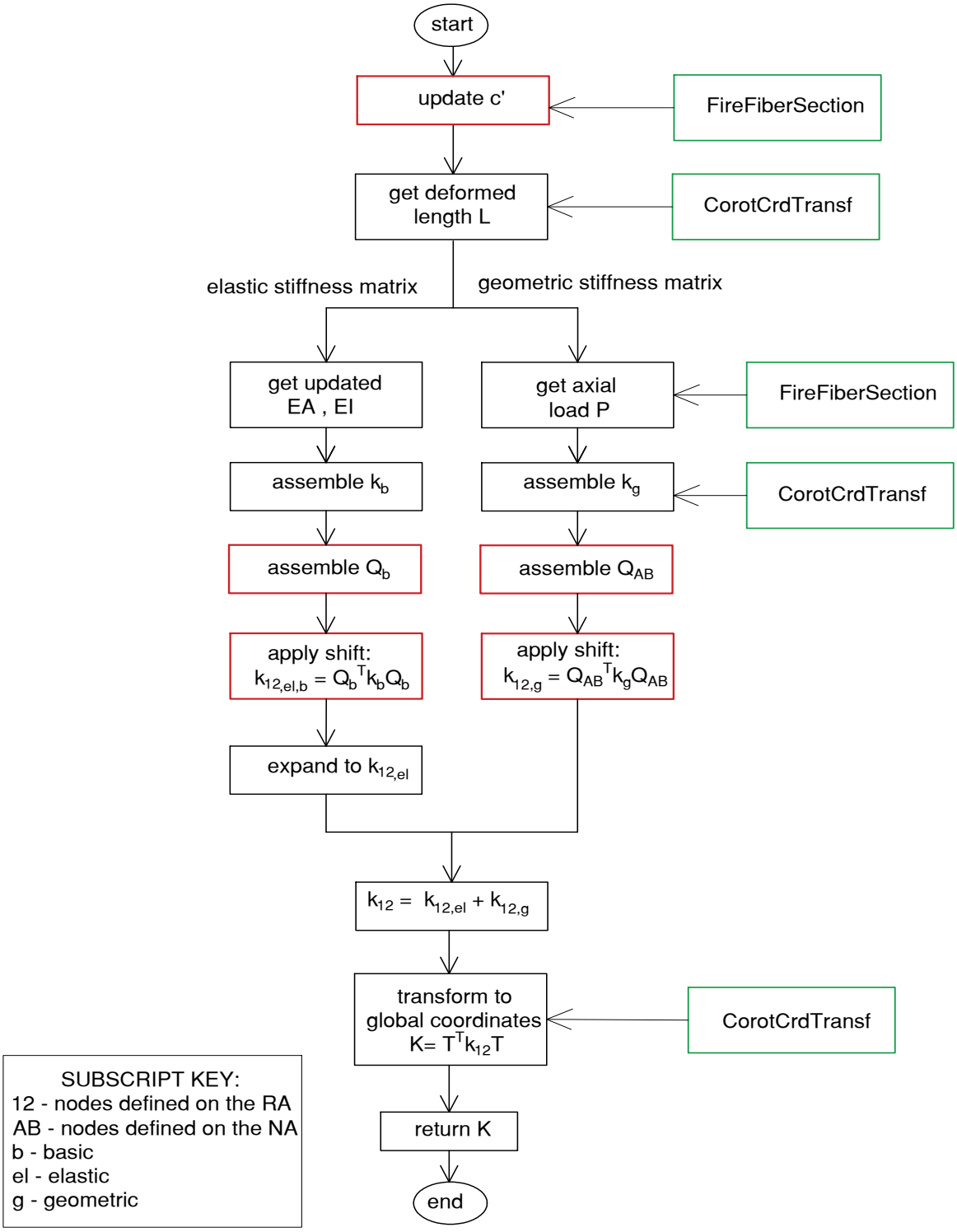


Figure 4.5: Steps of the function $getTangentStiff$ in class $FireEl$.

In OpenSees, the $[\boldsymbol{k_{el}}]$ matrix passed between functions is based upon a 3x3 matrix, instead of the typical beam stiffness matrix of size 6x6, as defined in Equation (3.6.5). This has the benefit that only the 3x3 matrix needs to be stored and used in computations, reducing storage space and calculation time. The compressed matrix is referred to as the basic stiffness matrix $[\boldsymbol{k_b}]$ and defined by:

$$[\boldsymbol{k_b}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{4EI}{L} & \frac{2EI}{L} \\ 0 & \frac{2EI}{L} & \frac{4EI}{L} \end{bmatrix} \tag{4.3.3}$$

Therefore, the matrix operator $[\boldsymbol{Q_{AB}}]$ as defined by Equation (3.6.10) also had to be adjusted to an equivalent 3x3 matrix displayed in Equation (4.3.4) and referred to as $[\boldsymbol{Q_b}]$. Note that this format is different to that presented in Section 3.6.2 due to the compilation of the matrix and the degrees of freedom (DOFs) considered, although carries out the same calculation.

$$[\boldsymbol{Q_b}] = \begin{bmatrix} 1 & c & -c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3.4}$$

The modified elastic stiffness matrix $[\boldsymbol{k_{12,el,b}}]$ in its basic format is therefore computed by:

$$[\boldsymbol{k_{12,el,b}}] = [\boldsymbol{Q_b}]^{\boldsymbol{T}}[\boldsymbol{k_b}][\boldsymbol{Q_b}] \tag{4.3.5}$$

The calculation of the geometric stiffness matrix starts by deriving the axial force P from the stresses of the *FireFiberSection*. Unlike the elastic stiffness matrix, the geometric stiffness matrix is kept as a 6x6 matrix and simply modified using the operator $[\boldsymbol{Q_{AB}}]$, as defined in Section 3.6.2.

After the elastic stiffness matrix is expanded back into the 6x6 matrix format by built-in functions of *CorotCrdTransf*, the local stiffness matrix can be determined as the combination of $[\boldsymbol{k_{12,el}}]$ and $[\boldsymbol{k_{12,g}}]$. The *CorotCrdTransf* function then converts the stiffness matrix into global coordinates to return the stiffness matrix $[\boldsymbol{K}]$ of the element.

4.3.3.3 Loading on the *FireEl*

As the NA shifts throughout the analysis, the load vector also requires updating in each iteration. In OpenSees, for each load-step the function *addLoad* generates the load vector for each element. This function was modified to account for the shift c' of NA derived from the previous iteration.

The *FireEl* has been developed to cater for three different types of loads, as listed below. The three types of loads and their equivalent nodal loads are illustrated in Figure 4.6.

1. Mechanical nodal loads in the form of a vector (referred to as $\{\boldsymbol{f_A}\}$) of size 3 and defined as in Equation (4.3.6).

$$\{\boldsymbol{f_A}\} = \begin{Bmatrix} F_{xA} \\ F_{yA} \\ M_{zA} \end{Bmatrix} \tag{4.3.6}$$

2. Mechanical beam loads applied as either a uniformly-distributed or a point load not applied at a node. These loads are converted to equivalent fixed-end forces at ends A and B. These are stored in a basic load vector format containing the axial force and both end moments, as shown in Equation (4.3.7).

$$\{\boldsymbol{f_b}\} = \begin{Bmatrix} N \\ M_A \\ M_B \end{Bmatrix} \tag{4.3.7}$$

3. Thermal beam loads applied as 25 temperature points along the height of the cross-section of a beam. The temperatures are first converted into thermal stresses and then into equivalent thermal loads N_θ and M_θ and stored in the basic load vector format:

$$\{\boldsymbol{f_{\theta,b}}\} = \begin{Bmatrix} N_\theta \\ M_{\theta,A} \\ M_{\theta,B} \end{Bmatrix} \tag{4.3.8}$$

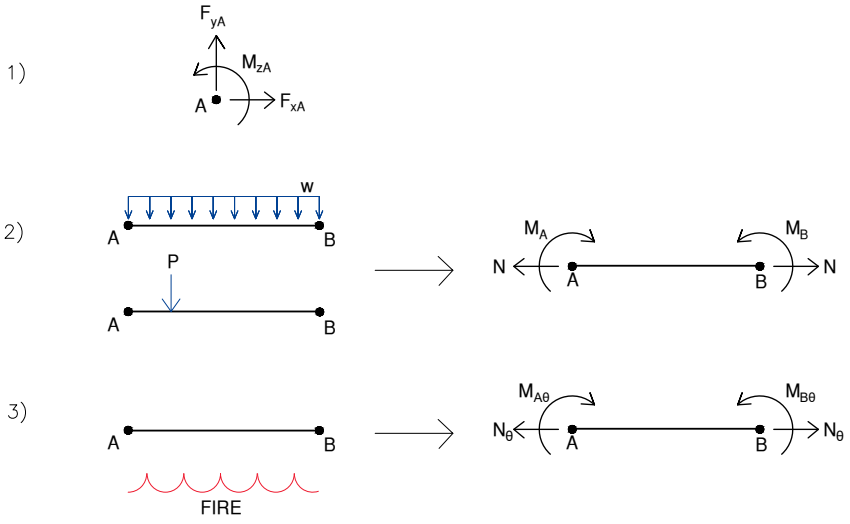


Figure 4.6: Load types applied to the *FireEl* and their equivalent nodal load format.

The function *addLoad* is called for each load-step and the steps are represented in the flowchart in Figure 4.7 and the actual C++ code is given in Appendix C.2.2.

The load-factor at a specific point in time and the details of the load are derived from the load class which is called. There are slight differences regarding how each loading type is processed, but in essence they all follow three steps:

1. multiply the load by the load-factor
2. convert to equivalent nodal loads, if not already in nodal format
3. apply the matrix operator to account for the shift in NA

The nodal mechanical loading is simply multiplied by a load factor and then modified by applying the transpose of $[\boldsymbol{Q}]$ as shown in Equation (4.3.9), which has been defined previously in Chapter 3 by Equation (3.6.7). The updated load vector $\{\boldsymbol{f_1}\}$ is returned.

$$\{\boldsymbol{f_1}\} = [\boldsymbol{Q}]^T\{\boldsymbol{f_A}\} \tag{4.3.9}$$

The mechanical beam loads are also multiplied by a load-factor but then require conversion into fixed end forces in the basic vector format. The NA shift c' is applied by the transpose of the basic shift operator $[\boldsymbol{Q_b}]$ (which has been previously defined in Equation (4.3.4)):

$$\{\boldsymbol{f_{12,b}}\} = [\boldsymbol{Q_b}]^T\{\boldsymbol{f_b}\} \tag{4.3.10}$$

The thermal fire load is represented by the 25 temperatures specified at a certain time by a time-temperature relationship that is given as an input file to the class *BeamThermalAction*. After converting the load into a basic vector of equivalent nodal loads, the shift is applied similar to the mechanical beam loads, as in Equation (4.3.11), and the updated thermal loads returned.

$$\{\boldsymbol{f_{\theta,12,b}}\} = [\boldsymbol{Q_b}]^T\{\boldsymbol{f_{\theta,b}}\} \tag{4.3.11}$$

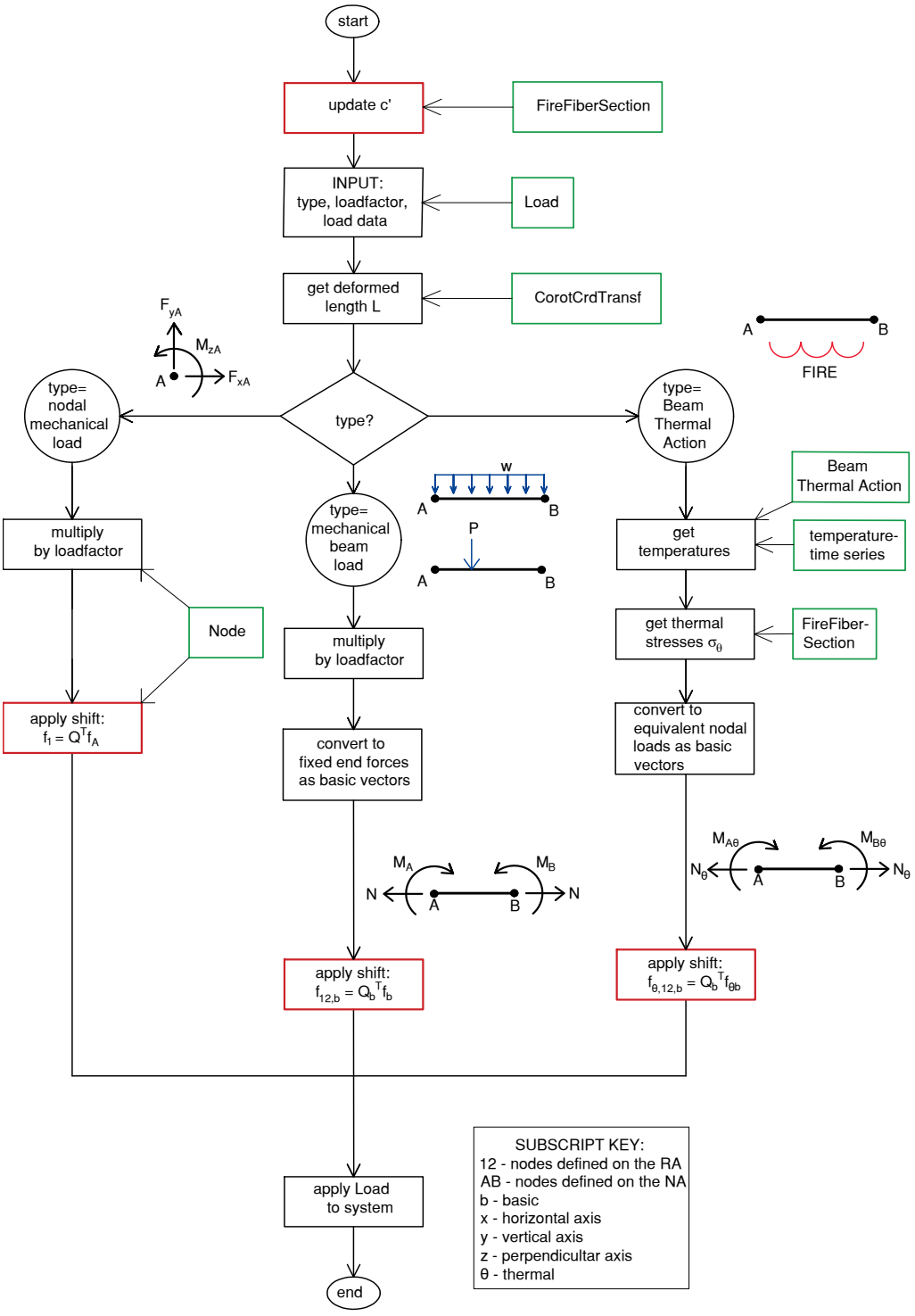


Figure 4.7: Steps of the function *addLoad* in class *FireEl* showing all three types of loading and how the shift is incorporated in each.

4.3.4 Implementation of FBE in the thermal-mechanical analyses in OpenSees

OpenSees for Fire has already implemented a thermal-mechanical analysis which performs the mechanical part of a decoupled analysis in fire. The process is incremental-iterative non-linear and can be categorised into three phases: predictor phase, corrector phase and convergence check.

The predictor phase calculates a displacement increment due to an initial unbalanced force. An unbalanced force develops in a non-linear analysis as a result of deformation and material softening. It is calculated as the difference between the applied force, thermal load and the resistance force. The corrector phase is an iterative procedure where the element force increment is calculated according to the predicted displacement increment. After each iteration, a convergence check is performed to see if the unbalanced force of each successive iteration is smaller than the tolerance specified. For the Newton-Raphson non-linear analysis, the load is applied in a series of steps and each step itself performs a number of iterations. The FBE methodology is integrated into these phases in OpenSees as shown in the flowchart in Figure 4.8, and will be explained in the following sections.

4.3.4.1 Inputs and load-steps

As with any structural analysis, the input required for the analysis is the geometry of the section and the structure, temperature-dependent material properties and the type, magnitude and direction of mechanical and thermal loads. For thermal loads, the input is in the form of temperature points specified along the height of a section.

4.3.4.2 Predictor phase

In each load-step i, the mechanical loads ($\{\boldsymbol{F_\sigma}\}$) and thermal loads ($\{\boldsymbol{F_\theta}\}$) are transformed into equivalent nodal loads. The shift in NA is considered by the modification of this load vector. The predictor phases calculates an initial unbalanced force $\{\boldsymbol{F_U}\}$ (in the form of an equivalent fixed end force) as a result of the thermal load and change of resisting force due to material softening ($\{\boldsymbol{F_r}\}$), as shown in Equation (4.3.12). To account for this material softening, the temperature-dependent material properties are updated at each step (Jiang and Usmani, 2013).

$$\{\boldsymbol{F_U}\} = \{\boldsymbol{F_\sigma}\} + \{\boldsymbol{F_\theta}\} - \{\boldsymbol{F_r}\} \tag{4.3.12}$$

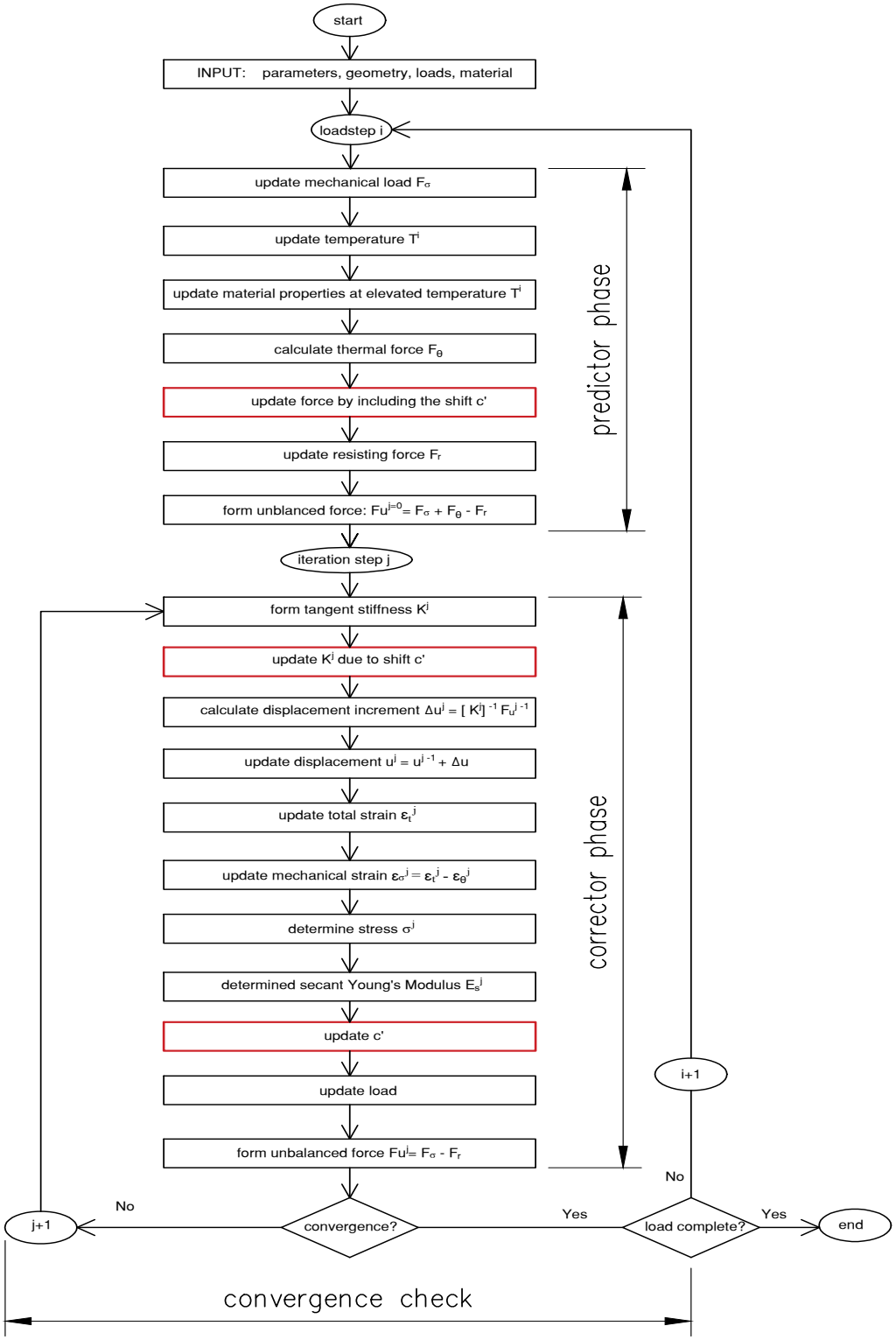


Figure 4.8: Thermal-mechanical analysis based on Jiang and Usmani (2013) with modification for the FBE shown in red

4.3.4.3 Corrector phase

The corrector phase determines the displacements as a result of the unbalance force. Firstly, the stiffness matrix from the previous converged step is updated to include the updated position of the NA (c'). Then the initial displacement increment ($\boldsymbol{\Delta u^j}$) of the iteration j is calculated by the unbalance force and the stiffness matrix at the previous converged step $j-1$:

$$\{\boldsymbol{\Delta u^j}\} = [\boldsymbol{K^{j-1}}]^{-1}\{\boldsymbol{F_U}\}^{j-1} \tag{4.3.13}$$

The displacement is updated, as well as the geometry of the structure. This is followed by the calculation of total strain, mechanical strain and stress, which provides a new secant Young's Modulus E_s and a new shift in NA c'. The resisting force is updated and a new unbalanced force can be determined:

$$\{\boldsymbol{F_U}\} = \{\boldsymbol{F_\sigma}\} + \{\boldsymbol{F_\theta}\} \tag{4.3.14}$$

For more details regarding this procedure refer to Jiang and Usmani (2013).

4.3.4.4 Convergence check

The last check of the analysis is the convergence check of the unbalanced force at the end of each iteration. Typically, the absolute value of the incremental unbalanced force has to fall below a tolerance value defined by the modeller. However, various other convergence test options are available in OpenSees, as specified in the command manual of OpenSees. If convergence is reached, the next load-step is analysed until the complete load has been applied. The tolerance used in this work was typically 1×10^{-5}.

4.4 Calling the FBE classes in OpenSees using TCL commands

Section 4.3 described how the C++ code was modified to account for the FBE methodology. However, the FBE classes and their in-built functions need to be made accessible to an OpenSees user with no access or understanding of the C++ code. As explained in Section 4.2.2, a user specifies the input to an OpenSees analysis model in the form of a TCL script. Therefore, the FBE classes need to be called by TCL commands. The TCL commands for creating instances of the classes developed for FBE are discussed in this section.

An instance of the class *FireFiberSection* with any number of fibres is built using the command:

FireFiberSection secTag {
fiber yLoc zLoc A matTag
fiber yLoc zLoc A matTag
...
}

where:

$secTag$	- the unique number of the section
$yLoc$	- location of the fibre on the y axis
$zLoc$	- location of the fibre on the z axis, typically zero for 2D
A	- area of the fibre
$matTag$	- number of the material created for the fibre

A *FireEl* beam class is created using the command:

element FireEl eleTag iNode jNode numIntegrPts secTag transfTag

where:

$eleTag$	- the unique number of the element
$iNode$	- first node number
$jNode$	- second node number
$numIntegrPts$	- not applicable for FBE but set as 3
$secTag$	- number of the FireFiberSection created for the beam
$transftag$	- number of the corotational transformation specified

All the other commands are as usually defined for OpenSees. For more details the command manual of TCL for OpenSees refer to Jiang (2017).

4.5 Conclusion

This chapter explained how the FBE formulation was implemented into the open-source FEA software OpenSees. OpenSees was chosen for this research as it is a powerful non-linear analysis tool and allows for the development of new elements and methods in its source code.

Firstly, the number of temperature inputs along the height of a cross-section has been increased from 9 to 25. However, the focus of this research was the development of classes *FireFiberSection* and *FireEl* written in C++ in Microsoft Visual Studio in order to integrate an intelligent beam element with a shifting NA into the framework of OpenSees. The validation of this implementation will be done using benchmarks and is discussed in the following chapter.

Lastly, TCL commands calling the FBE classes were developed so that the FBE methodology is available to users, who would typically write TCL input scripts for OpenSees analyses. Therefore, global 2D structures can now be modelled in OpenSees applying the FBE methodology by using the commands *FireFiberSection* and *FireEl*.

Chapter 5

Validation of the FBE model by numerical and experimental studies

5.1 Introduction

This chapter aims to validate the FBE implementation in OpenSees by applying it to case studies in literature. The results of the FBE analyses and experimental and/or numerical results are compared to evaluate how well the FBE methodology captures the structural response to fire. Three case studies in order of increasing complexity were chosen:

1. A benchmark of a simply-supported partially heated steel beam with an axial load from COST (2014). This study includes temperature gradients along the length of a beam, as well as high axial restraint loads.

2. Two tests of a simply-supported composite beam with a non-linear thermal gradient along the height of the section from Wainman and Kirby (1989). This case study has previously been considered using the first version of the FBE methodology in Prokon (version 0), and is included to compare results against existing predictions for validation, whilst also displaying how the FBE method is applied to composite steel-concrete cross-sections. Additionally, a discretisation study on the influence of the number of fibre elements on the predicted deflections was performed on one of the tests.

3. A benchmark of a steel frame based on one of the Cardington tests, which was subjected to non-uniform heating and axial restraint (Franssen *et al.*, 1995). This case study highlights how the simplified beam element developed is suitable for predicting deflections and axial forces even for structures where buckling occurs. It is shown that predicted results are significantly closer to experimental results than a contemporary beam element used in the literature, and deflections and axial forces are predicted with an accuracy comparable to more advanced modelling methods. Temperature is varied to highlight how results are affected by the temperature input values selected. Lastly, discreti-

sation studies are carried out, investigating the influence of the number of beam elements, and the number of fibre elements, on predicted deflection. In the previous version of the FBE research, the influence of (a) the number of beam elements along the length of a section, and (b) the number of fibres used to characterise a cross-section, has not been investigated. This work seeks to highlight the influence of such parameters on computational run time.

These models were built in OpenSees with the FBE formulation implemented as described in Chapter 4 and utilises the beam element *FireEl* with the *FireFiberSection* as a cross-section. The heating of the beam was modelled using the elemental load *Beam2DThermalAction* built into OpenSees by Jiang and Usmani (2013). For the FBE analysis of all three case studies, the Newton-Raphson method was chosen as the solution algorithm.

For the figures of case study structures included in this chapter, the numbers that are circled refer to the node numbers and the numbers that are blocked refer to the element numbers. All dimensions are given in millimetres. For clarity, the case study results developed in this research will be displayed in red in graphs.

5.2 Case Study 1: Partially heated steel beam

This fundamental case study is included to highlight how the FBE method captures the behavior of a beam with non-uniform heating along its length. The case study was one of four benchmarks that were created at the University of Sheffield in 2013 but based on experiments carried out by the University of Split (COST, 2014). The third test was chosen as it includes a vertical force as well as an axial force.

5.2.1 Description of benchmark

The experiment was a simply-supported steel beam with the dimensions as shown in Figure 5.1. The steel was classified as S355J2G3 according to EN 1993-1-2 (BSI, 2005) classification and tested to have mechanical properties of $f_y = 362.4$ MPa for the yield stress and $E_a = 209$ GPa for the Young's Modulus. The material model specified in this work, and previous research, is also based on EN 1993-1-2 (BSI, 2005), including stress-strain models, thermal strain model and reduction factors. The vertical (V) and horizontal loads (H) were specified as 200 kN and 400 kN respectively and were constant throughout the heating of the beam. A uniform temperature profile over the height of each cross-section was assumed. The beam was discretised into eleven nodes with ten elements, also depicted in Figure 5.1.

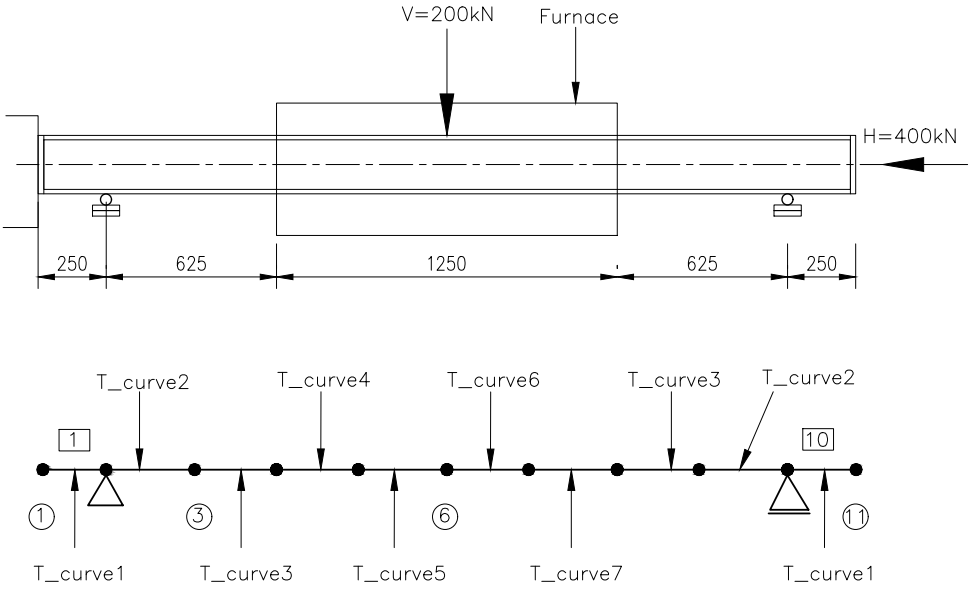


Figure 5.1: Simply-supported beam showing the loads and temperature curves for each beam element (reproduced from COST (2014))

Different experimental time-temperature curves were recorded at points along the length of the beam and allocated to each element. The outermost elements, element 1 and element 10, were subjected to the same time-temperature curve (T_curve1). Similarly, the first interior elements, element 2 and element 9, experienced the same heating (T_curve2). Therefore, seven different time-temperature curves were sufficient to describe the heating of the beam. For the detailed time-temperature curves, refer to Figure 5.2.

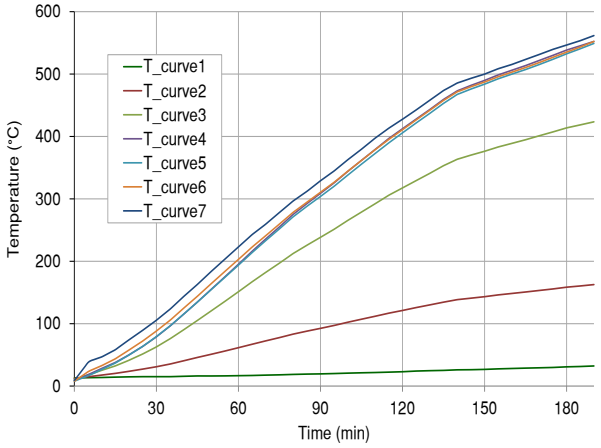


Figure 5.2: Time-temperatures curves recorded for the beam elements (COST, 2014)

The I beam section tested was a 212/180 profile with a flange and web thickness of 16 mm, as sketched in Figure 5.3. In the FBE analysis, the I-beam section was created using a *FireFiberSection* in OpenSees with 2 fibres along the height of each flange and 13 fibres along the height of the web. A finer mesh could have been used but it was found that more fibres did not affect the accuracy of the results in this case.

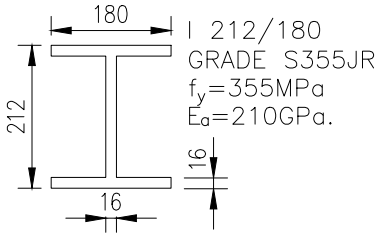


Figure 5.3: Steel profile of beam specified in COST (2014)

5.2.2 Results and discussion

To compare results, the vertical deflection at mid-span of the beam was measured throughout the analysis. In the COST (2014) project report, experimental deflections are provided, as well as a geometrically linear and geometrically non-linear analyses carried out using the software Vulcan (Burgess, 2015). The element used in Vulcan was a three-noded 2D beam element with six degrees of freedom per node and also adopts Euler-Bernoulli plane strain assumptions. The deflections predicted by Vulcan were plotted against the analysis deflection results of the FBE, as shown in Figure 5.4. The deflections calculated by the FBE formulation followed almost the same line to that of the Vulcan geometrically non-linear results, differing on average by 4% over time. This suggests that inclusion of the geometric matrix in the FBE methodology (refer to Chapter 3) is consistent with the numerical approach utilised by Vulcan.

At approximately 186 minutes, the FBE model does not converge, representing run-away failure. This is consistent with the experiment which shows a steep increase in deflection. However, all three of the numerical results underestimate the deflection recorded in the experiment. For example, the FBE deflections differs from the experiment by an average of 14% over time, typically under-predicting results. There are a variety of possible reasons for this, including Euler-Bernoulli plane strain assumptions being violated. This would lead to the FBE method underestimating deflection. Furthermore, issues that may affect the accuracy of predictions include how accurately the original steel temperatures were captured, the material models specified, how accurately the boundary conditions specified represeted those induced by the furnace, and similar aspects.

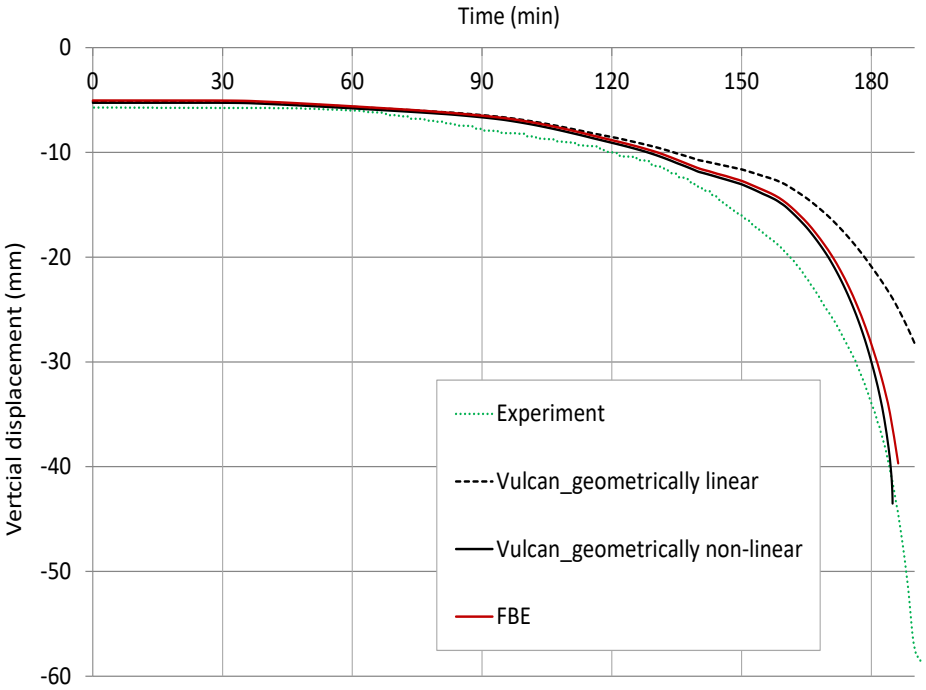


Figure 5.4: Vertical displacement at mid-span of the beam providing a comparison between the FBE model to the experimental results and Vulcan model results.

This simple case study has shown that the FBE method can capture behaviour such as large deflections and geometric non-linearity, while at the same time handling multiple temperature profiles. The results show good correlation to the Vulcan beam element used in the same benchmark. However, in this case, there is uniform heating over the height of the section, resulting in no change in NA position. It appears that the Vulcan formulation may not be able to capture shifting NA positions, although this is not clear from the literature. The shifting NA position of a composite section will be addressed in the next case study.

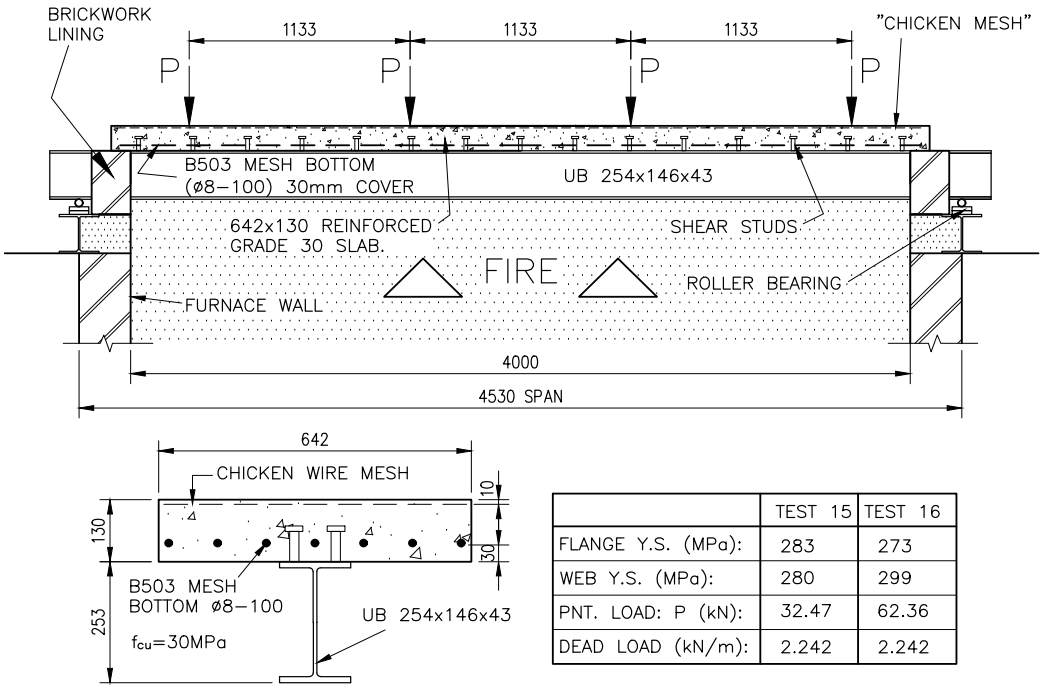
5.3 Case Study 2: Simply-supported composite beam

Two composite beam fire tests known as Test 15 and Test 16 (Wainman and Kirby, 1989) were chosen as the second case study for validation. The tests highlight the behaviour of composite beams when subjected to a non-uniform thermal gradient across the height of the section and have therefore become a well-known benchmark for validation of two-dimensional analyses of composite structures. Other researchers who have analysed this case study include Cedeno *et al.* (2011) and Benedetti and Mangoni (2007). More importantly, the case study has been separately used for validation by both Walls (2016) for the FBE method validation and by Jiang *et al.* (2014) for OpenSees. The method presented in this research (FBE version 1) in OpenSees will therefore be validated against the original FBE work (FBE version 0)

performed in Prokon Frame (Prokon SCL, 2015) - to ensure that the method was implemented correctly - and to the work of other researchers using OpenSees.

5.3.1 Description of the benchmark study

Both Test 15 and Test 16 followed the same experimental setup depicted in Figure 5.5, but the loading and the steel properties differed. The concrete grade was given as Grade 30, based on characteristic cube strengths. Therefore, using EN 1992-1-1 (BSI, 2004a), the average material parameters used were 33 MPa for the average compressive strength, 2.6 MPa for the average tensile strength and 31 GPa for the Young's Modulus. The B503 mesh used as rebar was specified as cold worked, high yield bar to BS4461 (Wainman and Kirby, 1989). The BS4461 code at the time suggested a characteristic strength of 460 MPa for bars that size, meaning that an average value of 520 MPa could be assumed for experimental work (i.e. two standard deviations in strength above the characteristic strength). The chicken wire mesh to prevent cracking in the top part of the slab was assumed to have negligible effect in the structural analysis and was therefore ignored.



	TEST 15	TEST 16
FLANGE Y.S. (MPa):	283	273
WEB Y.S. (MPa):	280	299
PNT. LOAD: P (kN):	32.47	62.36
DEAD LOAD (kN/m):	2.242	2.242

Figure 5.5: Experimental set-up for Test 15 and 16 reproduced from Wainman and Kirby (1989)

The temperatures recorded for the steel elements were provided as shown in Table 5.1 and in Figure 5.6 for Test 15, and in Table 5.2 and Figure 5.7 for Test 16. The furnace temperatures generally followed the ISO 834 test. A malfunctioning of the

instrumentation during the first 9 minutes of Test 15 (seen as the blank space in Table 5.1) resulted in a loss of data where temperatures had to be interpolated. Also note that the heat sink effect of the concrete reduced the temperatures of the top flange in comparison to the lower flange and web, where significantly higher temperatures were recorded. No data on the concrete temperatures was provided in literature.

Table 5.1: Temperature of steel elements recorded in Test 15

Temperature (°C)																
Time (min):	**0**	**3**	**6**	**9**	**12**	**15**	**18**	**21**	**24**	**27**	**30**	**33**	**35**	**36**	**39**	**40**
Upper flange					221	272	318	364	407	449	492	530	533	565	594	606
Web					473	540	592	632	657	679	700	718	728	731	745	749
Lower flange					455	535	598	644	674	697	716	729	737	741	756	762
Furnace					689	723	747	761	768	779	801	806	819	819	834	838
Std Fire curve	20	493	594	654	696	730	757	780	800	817	833	847	856	860	872	876

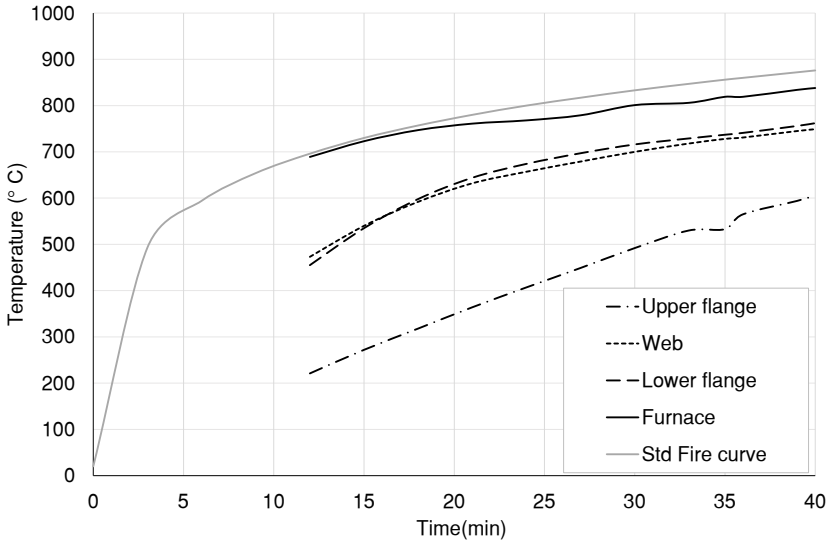


Figure 5.6: Temperature curves of steel elements for Test 15

Table 5.2: Temperature of steel elements recorded in Test 16

Temperature (°C)										
Time (min):	**0**	**3**	**6**	**9**	**12**	**15**	**18**	**21**	**22**	**23**
Upper flange	20	137	185	239	300	353	400	448	469	488
Web	20	183	285	388	476	542	590	625	636	647
Lower flange	20	153	248	359	462	544	602	642	654	666
Furnace	20	546	586	640	678	717	736	762	780	785
Std Fire curve	20	497	598	658	700	734	761	784	791	797

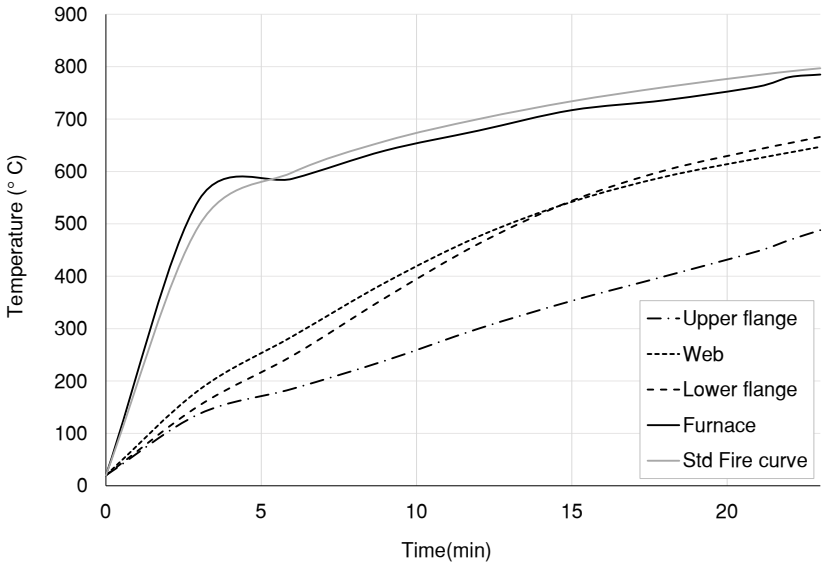


Figure 5.7: Temperature curves of steel elements for Test 16

5.3.2 Model using FBE method version 0

Walls (2016) modelled this benchmark using the FBE formulation by using Prokon Frame (Prokon SCL, 2015). The concrete temperatures were calculated using a temperature FEA model developed by Walls (2016) and based on EN 1992-1-2 (BSI, 2004*b*). Within Prokon Frame 16 beam elements were used along the length of the section. The shift in NA for each beam element was calculated and then implemented manually using rigid links. For clarity, the results of the previous FBE work will be referred to as FBE version 0.

The main difference between FBE version 0 and the model developed in this work (FBE version 1) is that the Prokon version can only consider statically determinate, simply-supported structures, while OpenSees can also consider statically indeterminate structures with restraint and structural continuity. Furthermore, the shift in NA is accounted for by rigid links in Prokon Frame, while matrix manipulation is used in OpenSees. This means that the NA is updated throughout the analysis in OpenSees, as opposed to updating it manually in Prokon. The core principles of the FBE formulation are the same for version 0 and version 1 and this case study is used to validate that the theory has been applied correctly.

5.3.3 Model in OpenSees using shells

Test 15 and Test 16 have been previously used for the validation of the structural fire engineering extension of OpenSees by Jiang *et al.* (2014). The 3D finite element model consisted of shell elements for the top concrete slab with beams elements for the steel beams, attached to the slab by rigid links. A typical OpenSees shell model is depicted in Figure 5.8. Note that the concrete ribs included in the figure

were not applicable for this specific case study, which included a flat slab. The shell elements include multiple layers which can consider temperature gradients. Heat transfer equations were used to determine these temperatures in the concrete. This modelling approach was therefore significantly more complex as opposed to the 2D FBE method developed in this research.

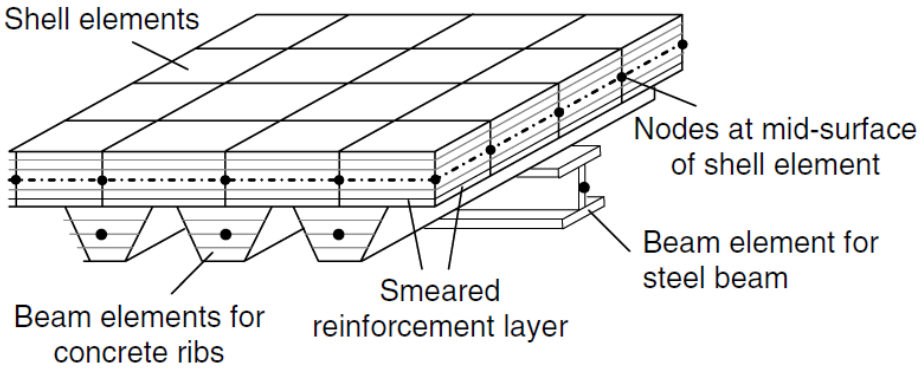


Figure 5.8: FE model in OpenSees using shells (Jiang *et al.*, 2014)

5.3.4 Model in OpenSees using FBE version 1

The model that was validated in this research applies the FBE model developed in OpenSees, as described in Chapter 4 of this document. The material models adopted were the inbuilt models in OpenSees based on the Eurocodes, specifically EN 1993-1-2 (BSI, 2005) for steel and EN 1992-1-2 (BSI, 2004*b*) for concrete. The beam consists of 38 *FireEl* elements, each 0.3 m long, as can be seen in Figure 5.9 a). The temperature was assumed to be uniform along each beam element length. The composite section was developed using a *FireFiberSection* with 8 fibres in the slab, a single fibre for the B503 mesh, 4 fibres in each steel flange and 8 fibres along the height of the steel web, as depicted in Figure 5.9 b). This fine discretisation was chosen in order to capture the non-uniform temperature gradient along the height of the section.

As described in Section 4.3.2.1, the existing OpenSees input capabilities were insufficient to capture the non-linear temperature distribution along the composite section. Therefore, OpenSees was modified to allow for the input of 25 temperature points along the height of a section. Two points were allocated each to the bottom flange, web and top flange. The remaining 19 points were therefore given at constant intervals along the height of the slab. The temperature points T1 to T25 are marked on the section in Figure 5.9 b). The temperatures for the concrete slab (T7 to T25)

were calculated by the temperature FEA developed by Walls (2016) and based upon the heat transfer equations of EN 1992-1-2 (BSI, 2004*b*), as in FBE version 0.

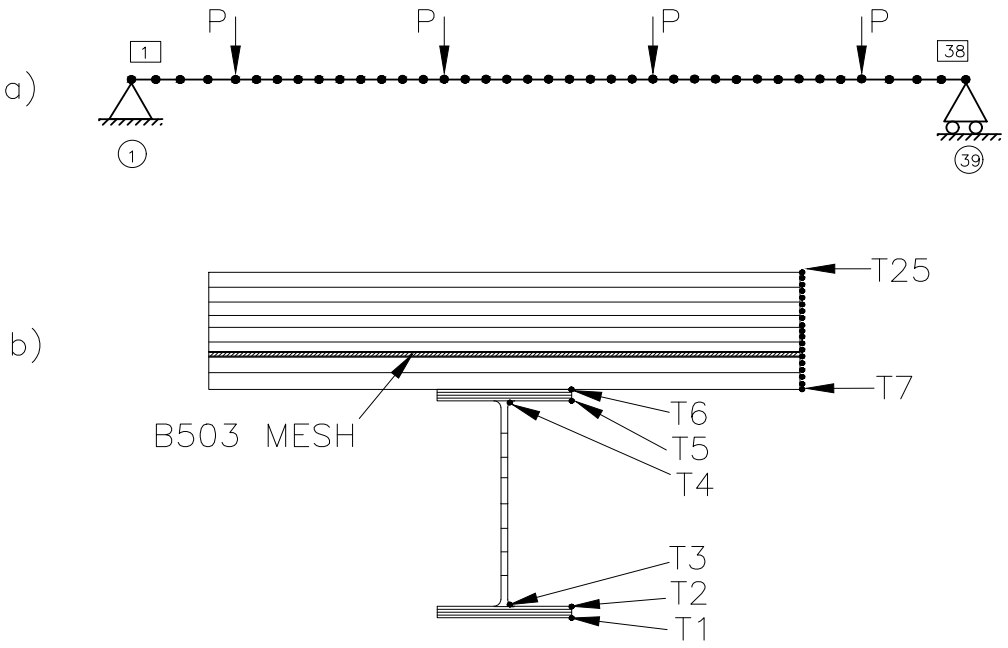


Figure 5.9: Diagram of beam showing the a) beam elements and b) the fibres of the section and the location of the temperature points T1 to T25.

5.3.5 Results and discussion

The deflection-time behavior at mid-span of the beams was measured in the experiments, and predicted by the FBE version 0, FBE version 1 and OpenSees analyses of both Test 15 and Test 16. Results are depicted in Figure 5.10 and in Figure 5.11 respectively. Both test results show good agreement between the FBE version 1 (red), the FBE version 0 by Walls (2016) and the OpenSees shell model by Jiang *et al.* (2014).

5.3.5.1 Test 15

The results of Test 15 are very similar for all three analyses. FBE version 1 follows almost the same trend as the FBE version 0, showing no more than 1% difference between predicted values. This highlights that in this work the FBE method has been correctly implemented in OpenSees. Furthermore, the FBE version 1 agrees well with the OpenSees shell model result by only showing a difference of 4%. Considering that the shell model has much higher numerical complexity, the FBE method is able to predict comparable deflections with much less user input complexity and computational power.

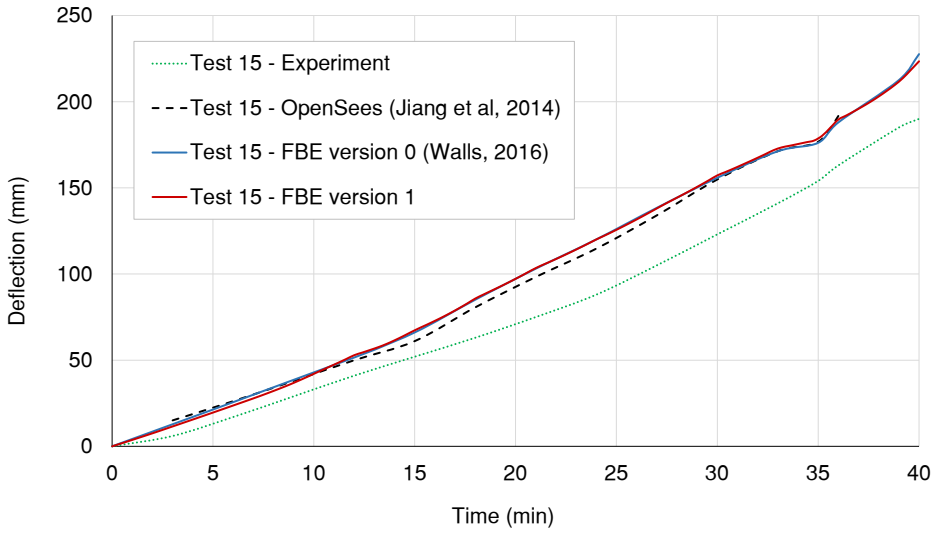


Figure 5.10: Deflection results for Test 15

5.3.5.2 Test 16

The predicted deflection curves for Test 16 also show general good agreement between the various numerical models. Again, FBE version 1 follows the same curve as FBE version 0 with an average difference of 1.4%. However, at failure, version 0 predicts a slightly higher deflection. This may be as a result of slight differences the use of the corotational approach used in OpenSees, which is not implemented in Prokon. FBE version 1 differs by 6% on average to the OpenSees shell model.

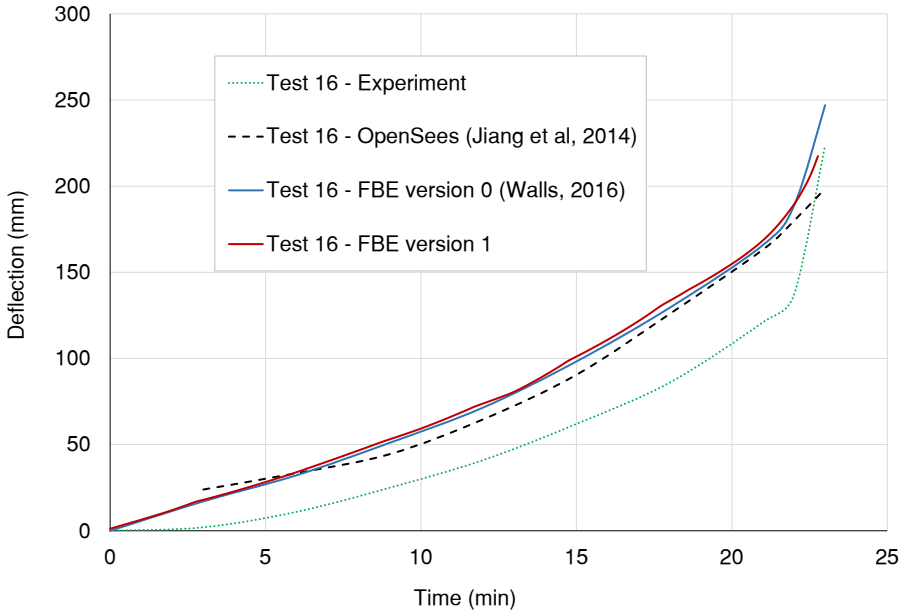


Figure 5.11: Deflection results for Test 16

5.3.5.3 Comparison to experimental results

For both tests, the deflections predicted by all three numerical models are higher than the experimental results. As already noted by Walls (2016), there are a variety of reasons that could account for this difference:

- The experimental uncertainty of temperature readings, especially due to the malfunctioning of equipment.
- Limited data on concrete properties was provided in literature.
- Slippage between steel and concrete may have occurred.
- Non-uniform heating along the beam could have occurred in the experiment.
- There was a lack of data for concrete temperatures.

Despite these differences and uncertainties, the overall trends of deflection were captured by the FBE model. This suggests that the FBE formulation can be successfully applied to composite beams with non-linear temperature profiles along the height of a section.

5.3.6 Fibre discretisation study

The effect of fibre discretisation of the composite cross-section on the results of a FBE analysis was investigated using Test 15 of this case study. Typically, a finer mesh would yield more accurate results. However, there may be a point when the mesh ceases to affect the results and a finer mesh would only increase computational effort without adding accuracy. This has not yet been investigated in FBE version 0 and, with the help of the implementation in OpenSees, FBE version 1 is used for parametric studies to consider this issue.

The fibre discretisation of the composite section as depicted in Figure 5.5 is of interest as it is not only subjected to a non-linear temperature distribution, but it is also a combination of three different materials. Test 15 was modelled in OpenSees with the number of fibres varied until no more change in deflection results could be detected, as seen in Figure 5.12. All the other parameters were kept constant as specified in Section 5.3.

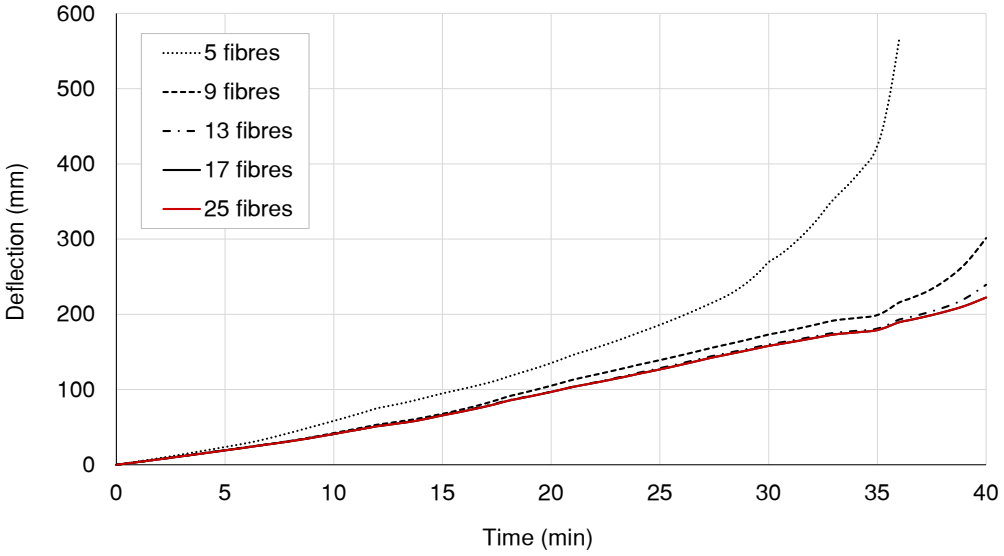


Figure 5.12: Deflection results of Test 15 using FBE models with different fibre discretisation. The result used for comparison in Section 5.3.4 is depicted in red.

The first model is very coarse mesh of 5 fibres: 1 fibre each for the concrete slab, B503 mesh, steel flanges and web. The results over-estimate the deflection and show run-away failure after 35 minutes. This could be due to the fact that a coarse mesh cannot fully capture the complex non-linear temperature distribution of the section, and the concrete cracking behaviour. The second model of 9 fibres has 2 fibres each in the flanges, web and concrete section, with again 1 for the B503 mesh. Increasing the number of fibres to 9 already improves the accuracy of the results, although the results still differ from the FBE model of 25 fibres by 36% at 40 minutes. The FBE models with 17 fibres (4 in each flange, web and slab with 1 for the B503 mesh) and 25 fibres (4 in each flange, 8 each in the web and slab and 1 for the B503 mesh) predict the same deflection, concluding that no more than 17 fibres are required for this analysis. This is still a surprisingly low number of fibres, suggesting that even a relatively coarse mesh in a composite section can yield accurate results. This will obviously depend on multiple factors such as geometry and the complexity of stress profiles developed.

5.3.6.1 Comparison in computational run time

The total computational run time taken by OpenSees for the loading, running and outputting of the FBE models with varying fibre number are represented in Table 5.3. Note that the models were run on a DELL Latitude E5570 with a Intel(R) Core(TM) i5-6300U CPU processor and 8.00 GB RAM. It is clear that the computational effort increases with the number of fibres used to define a cross-section, however, the increase is not significant. The total computational time consists both of (a) global analyses consisting of matrix method analyses, and (b) cross-sectional analyses

where the FBE formulation is implemented. Only part (b) experiences an increase in run time with increasing number of fibres, and this is a linear increase rather than a increase proportional to the degrees of freedom squared. Such a phenomena is beneficial as overall computational effort is not significantly affected with increasing computational complexity, making the analysis of large structures more feasible.

Table 5.3: Computational run time of the FBE models with different number of fibres along the height of the cross-section.

FBE model	Computational run time (s)
5 fibres	7.3
9 fibres	7.8
13 fibres	8.2
17 fibres	8.9
25 fibres	9.0

5.4 Case Study 3: Steel frame with structural continuity

For the third case study for validation, a benchmark based on one of the Cardington fire tests (Franssen *et al.*, 1995) was chosen. The test subject is a fully loaded, two dimensional steel frame and is significantly more complex than the two previous case studies. It combines non-uniform heating along the length of the beams and across the height of the cross-sections of beams. Furthermore, the structure is statically indeterminate and experiences axial restraint in the form of a spring. The case study is therefore a good validation for encompassing various effects such as non-uniform heating, thermal bowing, restraint, buckling and the consideration of stress distributions of structures in fire. COST (2014) included this benchmark for the validation of numerical models. It has been used to compare the predicted deflections and axial forces of 2D solid models built in CEFICOSS, SAFIR, ABAQUS and LS-DYNA and was therefore also chosen for validation of the FBE method in OpenSees developed in this research.

5.4.1 Numerical models of the benchmark

The frame had been previously analysed in literature by beam-column solid models in CEFICOSS, SAFIR and ABAQUS (as discussed in Section 2.7). The original model was built in CEFICOSS by Franssen *et al.* (1995) and the models in SAFIR and ABAQUS (COST, 2014) followed the same discretisation and mesh size. These models also performed heat transfer analyses of the frame and used the results as temperature inputs instead of the experimental temperatures measured in the test. In recent work by Rackauskaite *et al.* (2017), the LS-DYNA explicit dynamic solver has been validated against this benchmark. The steel frame was modelled using the Huges-Liu beam elements particularly developed to be simple and computationally efficient. This model therefore had a similar purpose to that of the FBE formulation

and offered a good comparison, as opposed to the models of ABAQUS, CEFICOSS and SAFIR.

5.4.2 Description of the test frame

The fire test was conducted by British Steel in collaboration with the Fire Research Station of the Department of the Environment (Franssen *et al.*, 1995) in Cardington. The set-up of the frame was designed to represent a typical office building. The steel frame consisted of a universal beam of 406 x 178 x 54 Grade 43A, which was bolted to two universal columns of 203 x 203 x 52 Grade 43A with dimensions as seen in Figure 5.13. The beam remained unprotected but loaded with four precast concrete slabs of 1200 x 5550 x 150 acting as the roof. There was a gap of 25 mm filled with ceramic fibre blanket between the beams and the slabs. This gap prevented composite action with the beam and it was assumed that the slab provided no structural stiffness. The columns were pinned at the base and extended above the beam. Fire protection for the webs of the columns was provided by aerated concrete blocks, but again the concrete added no structural support. The secondary steelwork around the frame offered some restraint and this is simulated by a bi-linear spring connected to the column. For further detail of the test refer to Franssen *et al.* (1995).

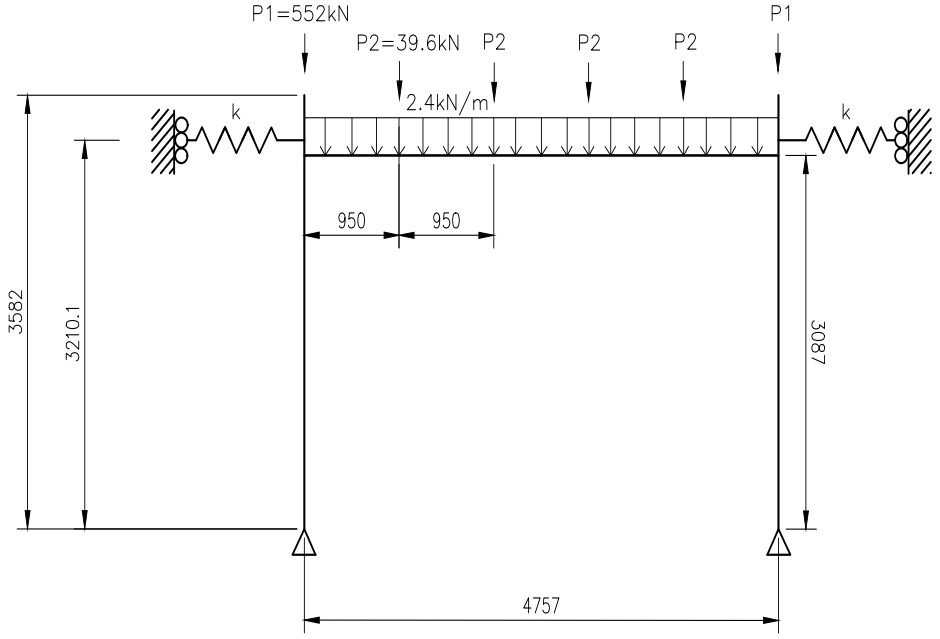


Figure 5.13: Schematic model of steel frame (reproduced from Franssen *et al.* (1995))

5.4.3 Loading

The mechanical loads applied were constant throughout the duration of the test. These loads included an axial compressive load of 552 kN applied to each column.

The beam was loaded with point loads of 39.6 kN each at four equally spaced positions along the span (Franssen *et al.*, 1995). The self-weight of the concrete was estimated to be a uniformly-distributed load of 2.4 kN/m (COST, 2014). After 22 minutes, the deflections exceeded span/32 and the mechanical load was removed for safety reasons.

5.4.3.1 Recorded temperatures

Throughout the duration of the fire in the compartment, the temperatures of the test frame were measured using thermocouples. The centre of the beam heated most rapidly, reaching temperatures of 775°C, 777°C and 577°C in the lower flange, web and upper flange respectively. The web temperatures recorded were higher than the flanges as heating occurs more rapidly in thinner members (Franssen *et al.*, 1995).

The column experienced lower temperatures than the beam. The inner facing flange of the column reached 606°C while the outer flange only reached 514°C, as result of the inner flange being subjected to a higher intensity of radiation than the outer flange. The column web was protected by the blockwork and therefore only experienced a maximum temperature of 250°C (Franssen *et al.*, 1995).

5.4.3.2 Modelled temperatures

Despite the availability of experimental data, the recorded temperatures had limited points in time and position. For example, the beam web only had one temperature reading in time. Therefore, the FBE model did not adopt the experimental temperature data for input but rather considered the temperatures from heat flow analyses performed in CEFICOSS, SAFIR and ABAQUS, which provide data for the entire experimental time.

The temperature curves resulting from these analyses and the experimental temperatures of the column are depicted in Figure 5.14. The column temperature is assumed to be uniform along the full height of the column. Note that the CEFICOSS data provided only limited points in time as opposed to the SAFIR and ABAQUS data. Also, the SAFIR temperature curve lies between the other temperture curves and can be regarded as a good average. For these two reasons, the FBE model adopted the SAFIR thermal analysis as temperature input for the analyses that follow in this section.

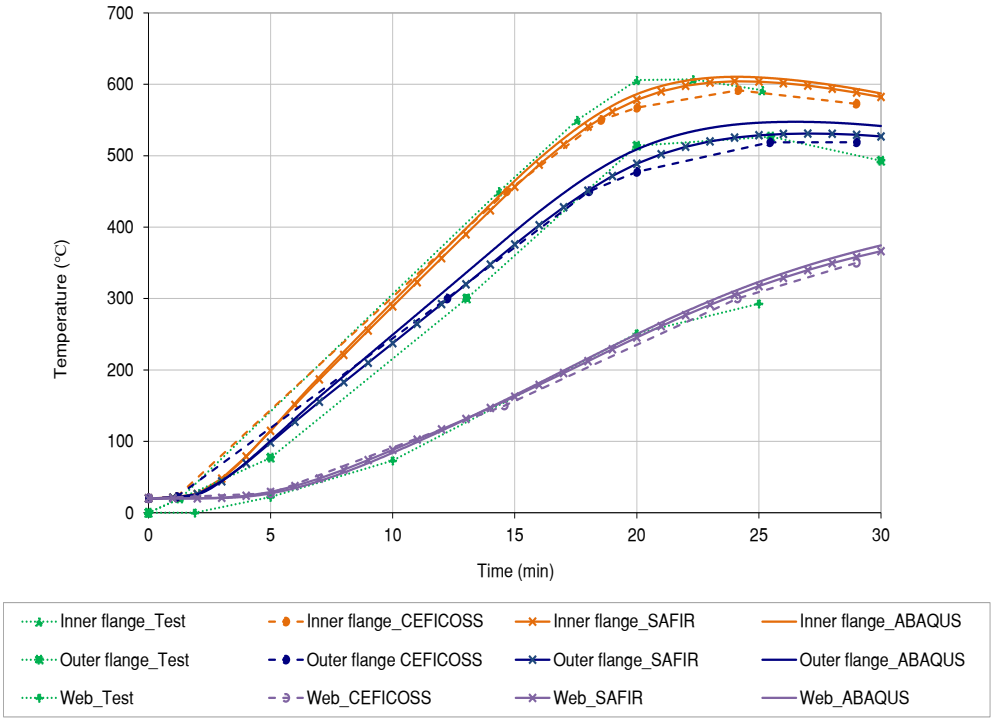


Figure 5.14: Temperature-time relationship of the column showing measured temperatures in the test (in dotted green) and those calculated by the heat-transfer analyses of CEFICOSS, SAFIR and ABAQUS provided by COST (2014).

Unlike the column, the beam experienced a non-uniform temperature distribution along its length. Closer to the connection, the beam temperatures recorded were slightly lower as a result of shielding and the heat sink effect of the secondary structure. The gas temperatures recorded in that area of the compartment were also lower than at mid-span of the beam. The temperature distribution along the beam was modelled using a reduction function $f(x)$. Franssen *et al.* (1995) specified a sinusoidal curve along the beam length x with a value of 0.9 at the connection and 1.0 at mid-span. The reduction function is defined by Equation (5.4.1) for the FBE model. To calculate the temperature $T(x)$ for each beam element, this function was multiplied by the beam temperature at mid-span $T_{mid-span}$, as seen in Equation (5.4.2). The value of f calculated for each element in the model of FBE is given in Table 5.4. The value for x was taken at the mid-point of each element and f was considered uniform along the element.

$$f(x) = 0.9 + 0.1 \times sin(0.21\pi x) \tag{5.4.1}$$

$$T(x) = f(x) \times T_{mid-span} \tag{5.4.2}$$

Table 5.4: Reduction function of temperature for beam elements

Element no.	13	14	15	16	17	18	19	20
x **(m)**	0.317	0.633	0.950	1.267	1.583	1.900	2.139	2.378
f	0.910	0.931	0.950	0.967	0.981	0.991	0.997	1.000

As for the column, the mid-span temperatures were taken from the SAFIR thermal analysis as depicted in Figure 5.15. Again, the SAFIR analysis appear to represent a good average between the experiment, and the CEFICOSS and ABAQUS thermal analyses, as well as providing sufficient data points in time.

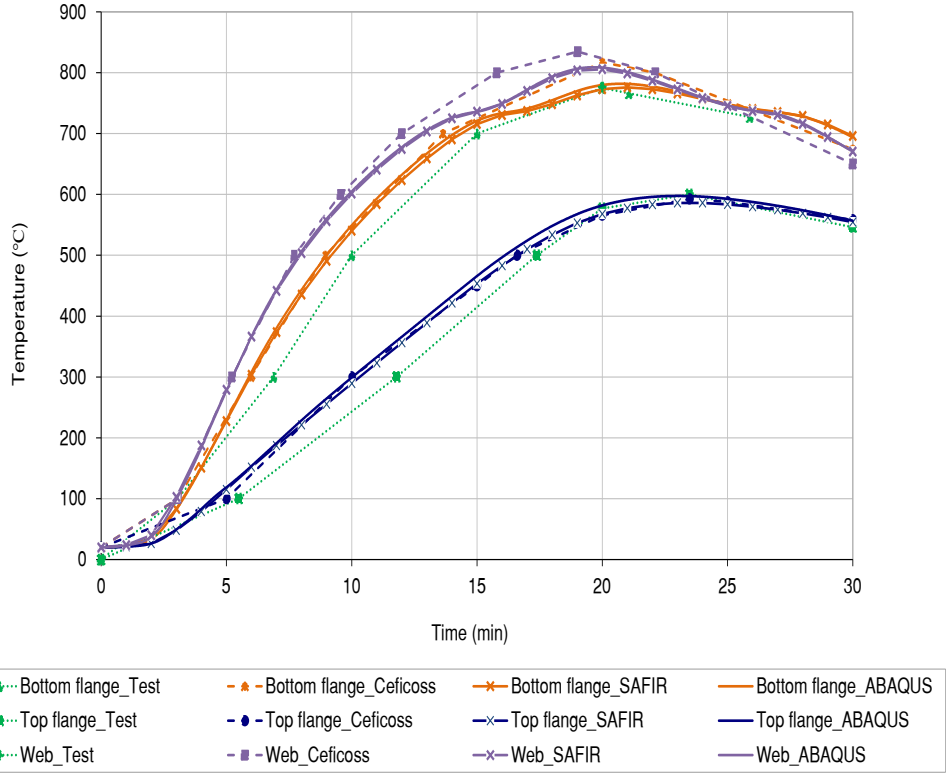


Figure 5.15: Temperature-time relationship of the beam at mid-span showing measured temperatures in the test (in dotted green) and those calculated by the heat-transfer analyses of CEFICOSS, SAFIR and ABAQUS provided by COST (2014).

5.4.4 Modelling the frame for mechanical analysis

The frame discretisation specified by Franssen *et al.* (1995) is depicted by Figure 5.16. Symmetry allows the analysis of only half the frame and symmetric boundary conditions are applied at the midspan of the beam. As the concrete offers no structural resistance and only thermal boundary conditions and loading, only the steel is modelled using 2D elements. A rigid connection between the column and beam was

assumed, although this is not always the case in practice. However, the temperature at the connection remained lower than in other parts of the steel frame, and a connection with sufficient rigidity was provided (COST, 2014), allowing for this assumption to be valid. For the FBE analysis, all column and beam elements were modelled by a total of 17 *FireEl* elements, 9 for the column and 8 for the beam. The column and beam profiles were modelled using a *FireFiberSection* each with 2 fibres along the height of each flange and 13 fibres along the height of the web. The number of fibres required will be further discussed in Section 5.4.7.

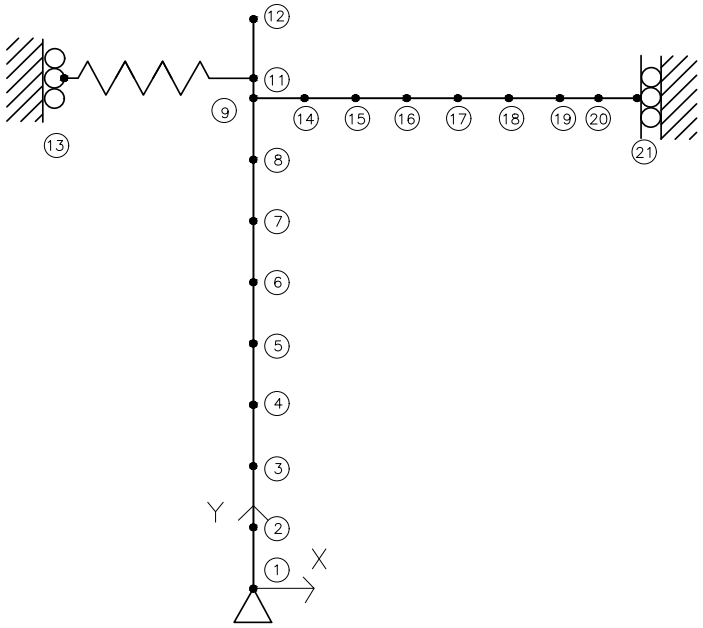


Figure 5.16: Discretisation of frame, reproduced from Franssen *et al.* (1995).

The has a force-displacement behaviour of the bi-linear spring simulating the secondary steelwork is shown in Figure 5.17. The axial stiffness was calculated by Franssen *et al.* (1995) as 6700 kN/m with an axial plastic load of 86 kN. The material properties and reduction factors of the steel at high temperature were based on EN 1993-1-2 (BSI, 2005). A hardness test had been performed at the time of the experiment which confirmed that the steel satisfied the nominal tensile strength for BS 4360 Grade 43A, however, the actual yield strength of the steel was never tested. Therefore, Franssen *et al.* (1995) performed five numerical simulations of the test using different values of yield strength and concluded that an ambient yield strength f_y of 408 MPa with an ambient Young's Modulus E_a of 210 GPa yielded the best fit to the deflections of the experiment. The thermal expansion coefficient was specified to be constant at 0.000014 /°C by COST (2014).

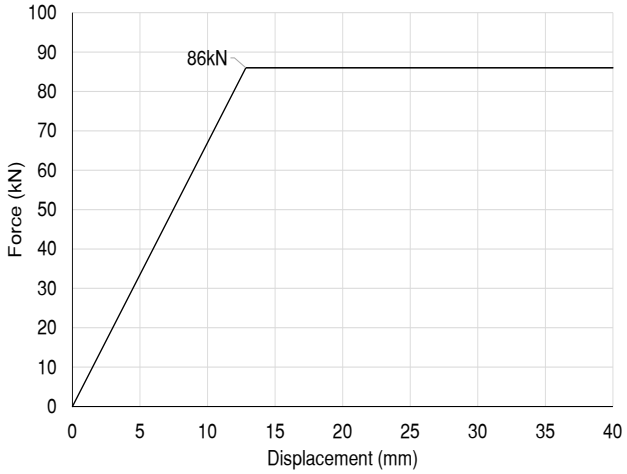


Figure 5.17: Spring behaviour acting as the restraint provided by the secondary structure (reproduced from COST (2014))

5.4.5 Results and discussion

The FBE model was validated by comparing beam mid-span deflection, column lateral displacements and beam axial forces to the experimental results as well as to the numerical results of the various FE models. For all three cases, the FBE analysis results show good correlation to the general trends and behaviour of the frame. Note that the analysis ceases to converge at approximately 19 minutes, representing run-away failure. This is approximately two minutes before the deflection in the experiment exceeded span/32 and the test was stopped.

5.4.5.1 Beam deflection

The mid-span deflection of the beam, as well as the rate of deflection, increased with the temperature rise. The comparison of the deflection of the beam at mid-span is represented in Figure 5.18. It appears that the FBE deflections (red) at mid-span follows a line closest to the SAFIR results, as expected since the models adopted the same temperature inputs. At the 15 minutes mark, the FBE results differ in relation to SAFIR by 6% and under-estimated the experimental results (dotted green) by 11%. Run-away failure and large deflections is predicted earlier by approximately 2 minutes by the FBE than was recorded during the experiment.

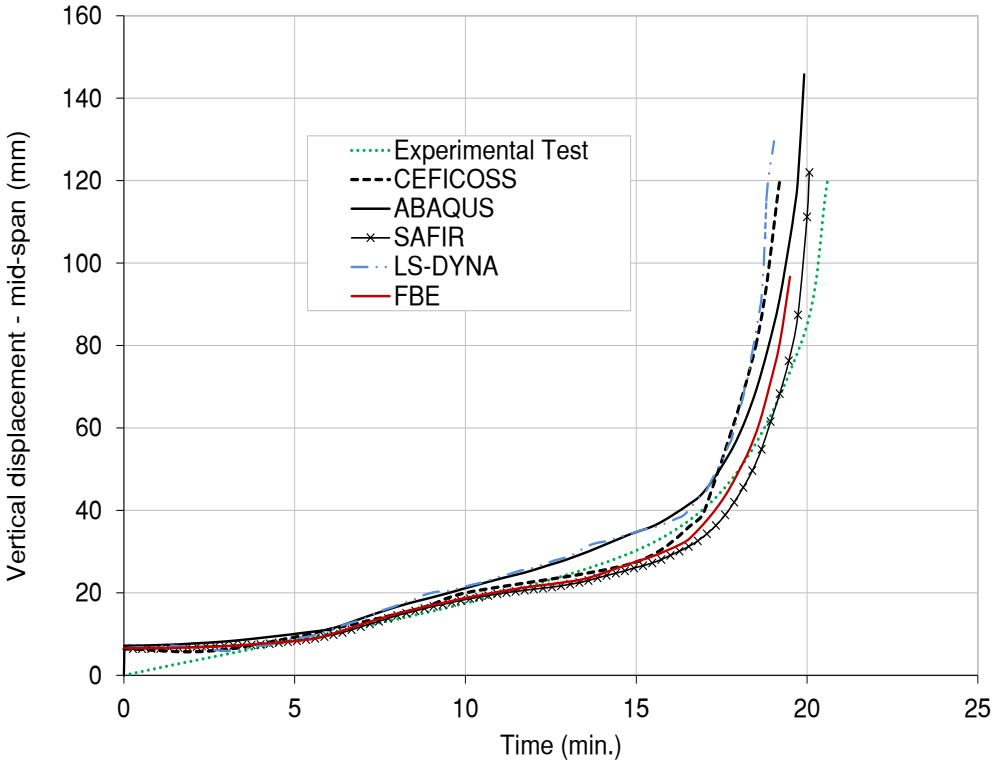


Figure 5.18: Deflection at mid-span of the beam

5.4.5.2 Column displacement

The column experienced lateral displacement to the left due to the axial expansion of the beam and the rotation of the end of the beam as a result of thermal bowing. The results of this lateral displacement at mid-height are depicted in Figure 5.19. No experimental or LS-DYNA displacement-time data was available for this comparison. Initially, the FBE curve follows a pattern similar to the ABAQUS results, differing at most by 10%. At 15 minutes, the FBE results differed to the other numerical models by no more than 17%. Without having experimental data available it is difficult to determine which model provides the most accurate prediction, although general trends are comparable between all four models.

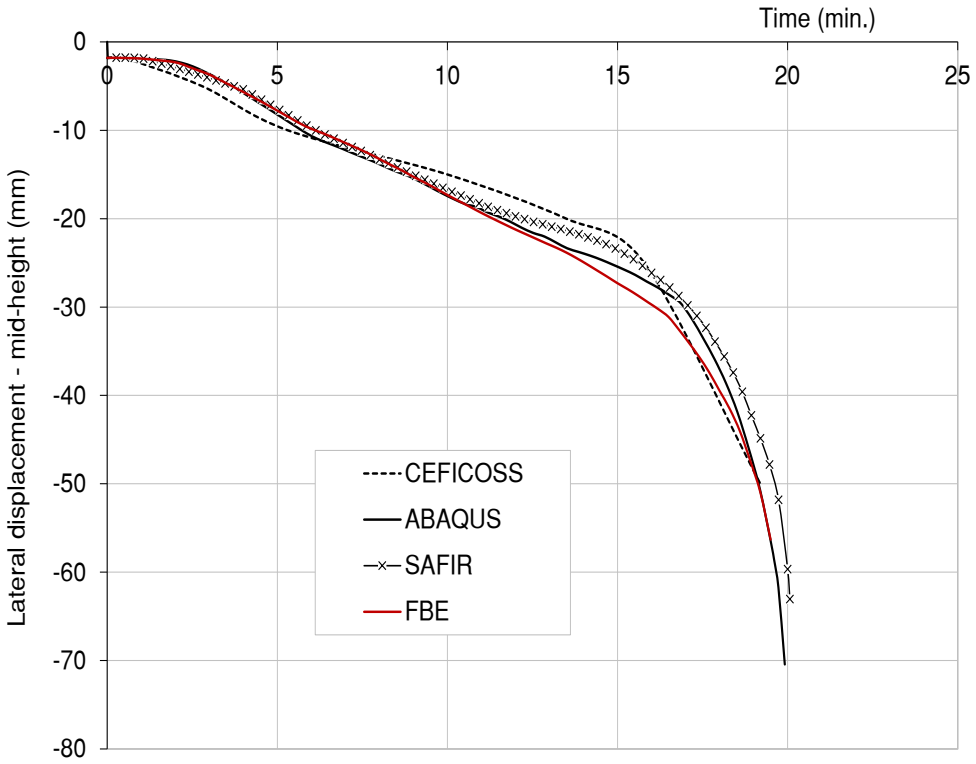


Figure 5.19: Lateral displacement at mid-height of the column

5.4.5.3 Axial force in beam

For the last comparison, the axial force in the beam is numerically predicted, as displayed in Figure 5.20. Again, no experimental data was provided. A good agreement between FBE (red), CEFICOSS, ABAQUS and SAFIR models can be seen with a steep increase in compressive force with a peak between 11 and 12 minutes. The peak force of the FBE model is reached at similar times as the other numerical models, with the predicted peak of FBE being between 3.4% and 5.5% lower, and axial forces reduce more gradually after buckling occurs.

It is interesting to note that the FBE axial force displays a much closer agreement to the other models than the LS-DYNA axial force result (blue), considering that both models use beam elements. At 15 minutes the FBE model differs by no more than 4% to the other models, whereas the LS-DYNA model shows a difference of more than 23% at the same time point. The LS-DYNA force does not predict a peak, followed by buckling and then a reduction in force, as is seen in the other models. Although both these methods used simple beam formulations, the FBE beam elements appears to capture the behaviour more fully. It appears that the updating of the neutral axis position, and associated calculated stiffnesses, have a significant influence on the results obtained.

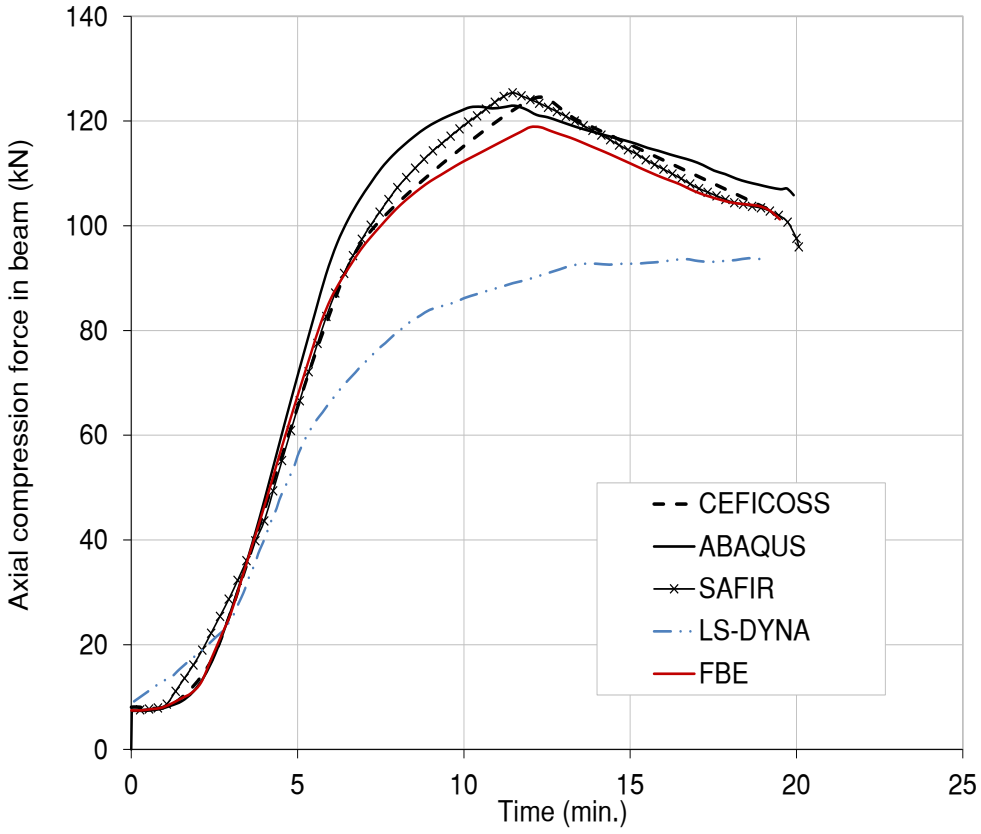


Figure 5.20: Axial compression force in beam

5.4.5.4 Comparing model and experimental results

There can be a number of reasons why the results of the FBE model and the other numerical models show certain differences. Firstly, the temperature inputs of the experiment and the numerical models vary, especially as the ABAQUS, SAFIR and CEFICOSS models used their own thermal analyses as the input, meaning that an exact comparison is not possible. The sinusoidal reduction function along the length of the beam applied to each beam element could be interpreted differently by each modeller. Secondly, the built-in material models of the software may have slight differences. Lastly, if Euler-Bernoulli assumptions were violated due to local distortions it may have influenced results.

Despite these differences, the FBE results display a good correlation to the experiment and the numerical models. This suggests that the FBE formulation was able to capture critical structural behaviour of a complex fire scenario with structural continuity, buckling and non-uniform heating patterns.

5.4.6 Temperature input sensitivity study

In a real fire scenario, temperatures are difficult to predict (as explained in Section 2.4.1) and therefore the modelling of a structural response to fire has high inherent uncertainty regarding the temperature inputs. The sensitivity of the FBE model to temperature inputs have been tested by both increasing and decreasing all the temperature inputs for the steel frame by 20%. The predicted deflection, displacement and axial compressive force are compared to the results in the previous section (shown in red) in Figure 5.21 to 5.23. By utilising a simple model that can produce results quickly, sensitivity analyses can be more easily carried out. In design scenarios, where input parameters are not accurately known, it may be more important to understand the sensitivity of a model to input parameters and to compare relative behaviour of alternative designs, rather than seeking to produce highly accurate results.

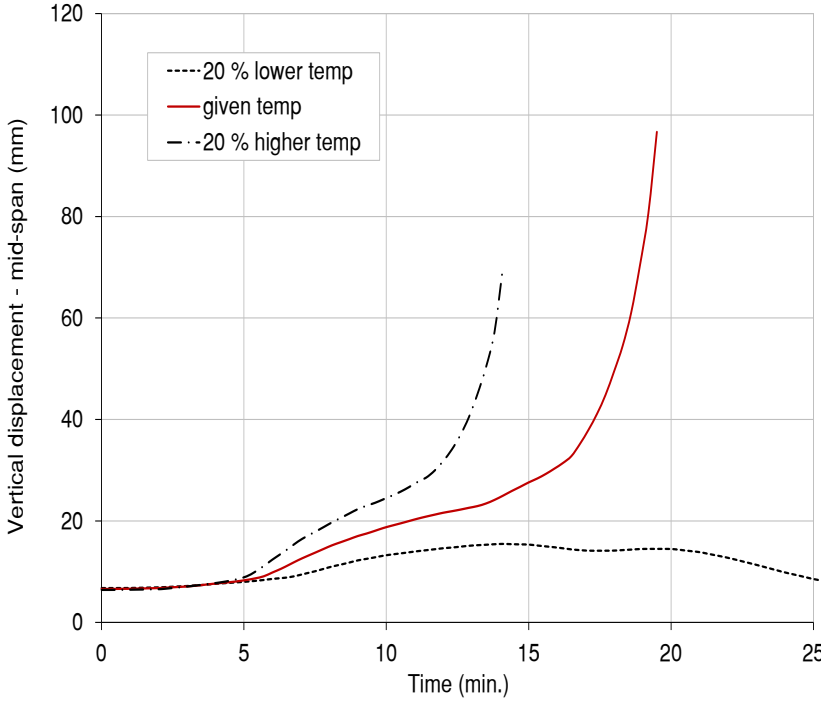


Figure 5.21: Vertical deflection at mid-span of beam using FBE models with temperature inputs 20% lower and 20% higher.

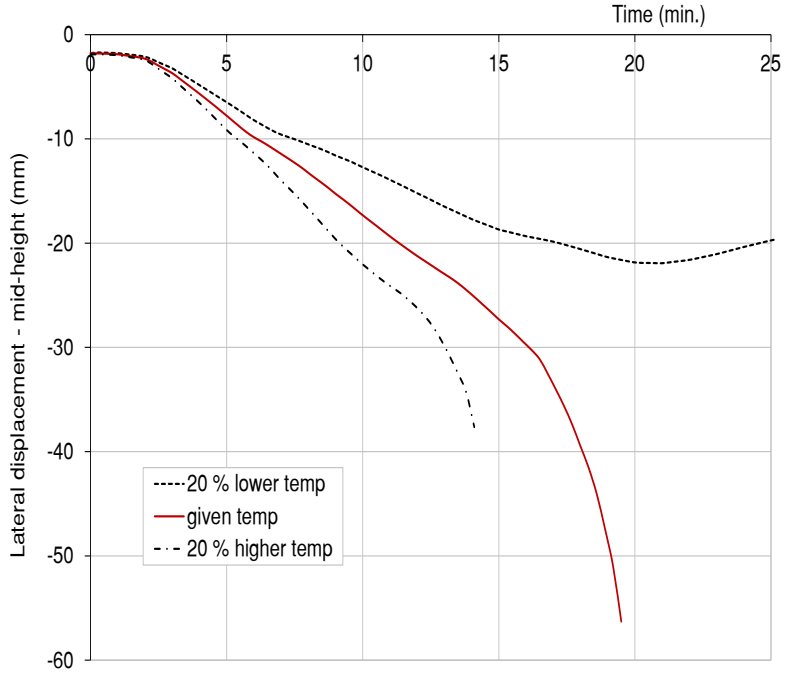


Figure 5.22: Horizontal deflection at mid-height of column using FBE models with temperature inputs 20% lower and 20% higher.

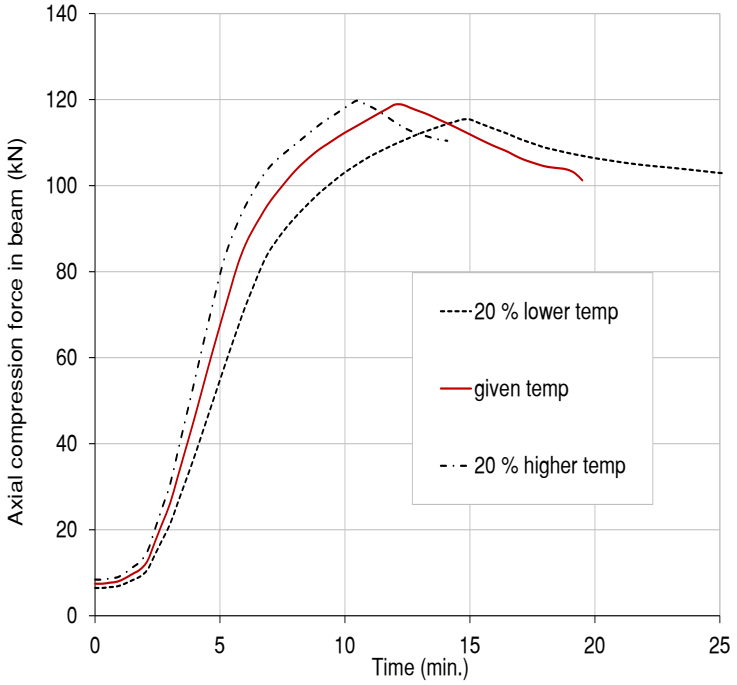


Figure 5.23: Axial compression force in beam using FBE models with temperature inputs 20% lower and 20% higher.

The results clearly show that a 20% temperature difference affects the behaviour of the frame significantly. The higher temperature model predicts failure at 14 minutes as opposed to at 19 minutes predicted by the given temperature inputs. The peak axial force predicted in the beam with 20% higher temperatures is less than 1% higher than in the original FBE model (red line).

In comparison, the model experiencing 20% lower temperature predicts no failure at all. The results show a recovery after 20 minutes, where the temperature decreased and the material contracts so that deflection, displacement and axial force decrease accordingly. The peak compressive force is predicted at 15 minutes and approximately 3% lower than the FBE model (red line) with given temperatures, as seen in Figure 5.23.

This study highlights that, similar to other thermal-mechanical models, the FBE model is sensitive to temperature inputs. A slight decrease or increase in temperature could affect the structural response and failure time. Therefore, for analysing structures in fire, the accuracy of the temperature data is of great importance. The benefit of using a simple model is that many analyses with different temperature profiles can be tested with minimal additional computation time and effort.

5.4.7 Discretisation study

A discretisation study was performed on the steel frame to investigate the effect of mesh size on the FBE model. This was done by increasing the number of fibres and elements respectively until no more change in results could be detected. As in most FEA models, accuracy is expected to increase with a finer mesh. However, a too fine mesh typically increases computational time and modelling effort without adding accuracy or changing the results.

5.4.7.1 Element discretisation

The optimum number of elements for the steel frame in CEFICOSS has already been investigated by Franssen *et al.* (1995) and found to be 18 elements. Therefore, the discretisation as depicted in Figure 5.16 has become the standard for this benchmark (COST, 2014) and used in the FBE model for comparison to other models in the previous section. Nevertheless, the influence of the number of elements for the FBE model was still of interest. Therefore, the FBE model was rerun using a coarser and a finer mesh.

The coarser mesh consisted of a total of 9 elements (half the original number of elements), with 6 for the column and 3 for the beam. The finer mesh had a total element number of 35 (almost double the original number of elements) with 19 for the column and 16 for the beam. All the other inputs were kept constant.

In general, the number of elements does not seem to have a significant effect in this case. The mid-point column displacement and the axial force in the beam for the models with 9, 18 ad 35 elements were almost the same. The comparison of the

predicted mid-span beam deflection of the three FBE models with different element numbers can be seen in Figure 5.24. The FBE models with 18 and 35 elements results show almost no difference in the predicted deflection, except that the finer mesh could predict deflections closer to the failure point a few seconds after the 18 element model (as seen by the extended curve of the 35 element model).

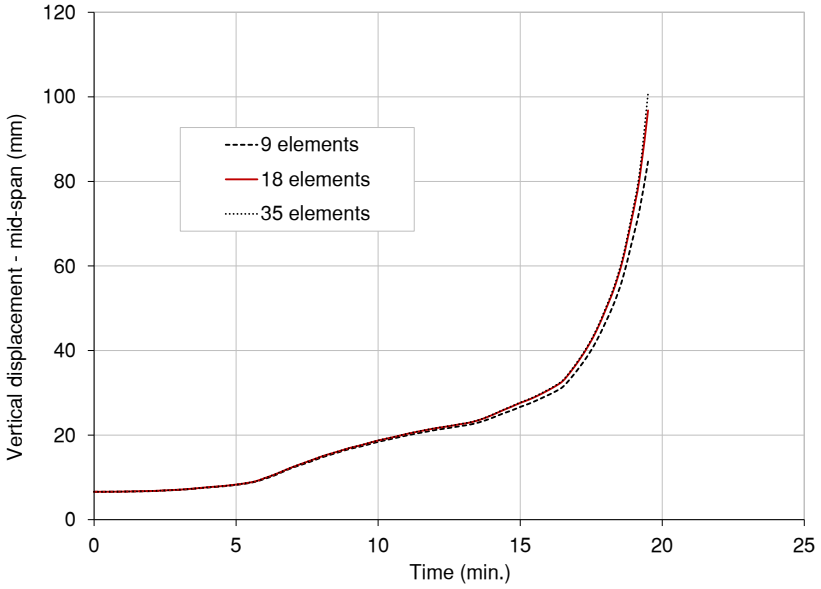


Figure 5.24: Vertical deflection at mid-span of the beam predicted by FBE models with different element discretisation. The results used in Case Study 3 are depicted in red.

It appears that the more coarse the mesh, the sooner the model cannot obtain cross-sectional convergence, resulting in run-away failure. The 9 element model predicts slightly lower deflections. However, this is probably as a result of the non-linear temperature distribution along the length of the beam. The temperature along a single beam element is assumed to be constant. Therefore, with fewer elements, the thermal gradient along the beam length can not be modelled as accurately as in the models with more elements.

The computational run time of the different models are compared in Table 5.5. As expected, the run time increases with the number of elements. The computational effort is significantly greater for 35 elements at 12.8 s than for 18 elements at 5.1 s, supporting the result of Franssen *et al.* (1995) that a 18 element mesh is more efficient. In the case of larger structures, a coarse mesh may therefore be more suitable, as a finer mesh would require a significantly greater computational effort, although this would depend on various factors.

Table 5.5: Computational run time of the FBE models with different number of elements

FBE model	**Computational run time (s)**
9 elements	2.8
18 elements	5.6
35 elements	12.8

5.4.7.2 Fibre discretisation

To investigate the effect of the number of fibres specified in an analysis, a similar study as described in Section 5.3.6 was applied to the steel frame. A coarse FBE section consisted of only 3 fibres, one for each of the flanges and the web. This was increased to 7 fibres, 2 in each flange and 3 in the web. The fibre number was then increased to 17 and ultimately 25 fibres: 4 in each flange and 17 in the web. All other parameters were kept constant as specified in Section 5.4.4.

The predicted deflections at mid-span of the beam by FBE models with various number of fibres can be seen in Figure 5.25. The coarse mesh of 3 fibres predicts a deflection approximately 31% higher than the other models. By increasing the number of fibres only slightly to 7 fibres, the accuracy is greatly improved and the predicted deflections differed by only 2% on average. The deflections predicted from the models with 17 and 25 fibres are almost indistinguishable, suggesting that a finer mesh than 17 fibres is not required. The mid-span deflections and the axial force in the beam are not shown, but they follow similar trends as the vertical deflections.

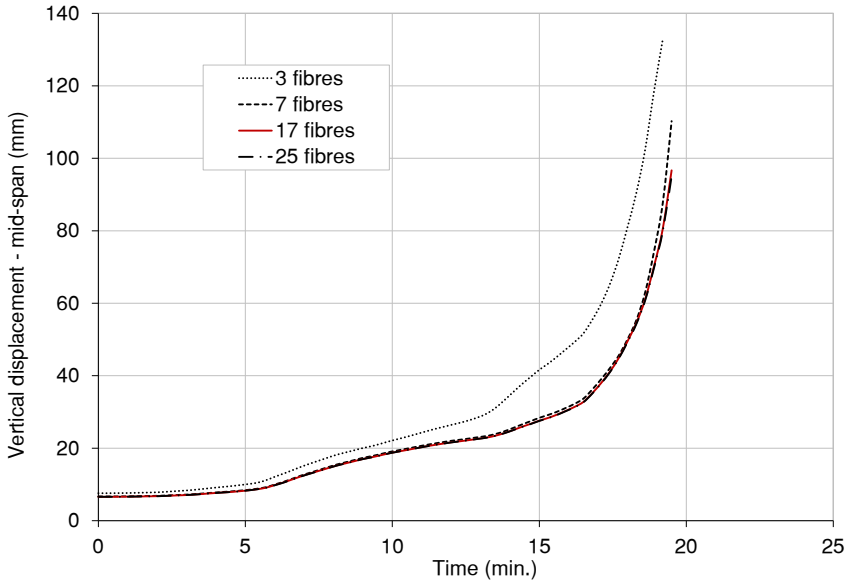


Figure 5.25: Vertical deflection at mid-span of the beam using FBE models with different fibre discretisation. The results used in Case Study 3 are depicted in red.

The computational run time of the FBE models with different number of fibres are compared in Table 5.6. Again, the run time increases as the mesh gets increasingly fine, but not as significantly as in the element discretisation study. This suggests that for larger structural models, the effect of the element mesh is more significant to the run time of a model.

Table 5.6: Computational run time of the FBE models with different number of fibres in cross-section

FBE model	**Computational run time (s)**
3 fibres	2.7
7 fibres	3.6
17 fibres	5.1
25 fibres	5.6

5.4.7.3 Conclusion of discretisation study

The discretisation study of the steel frame reveals that the FBE analysis requires less elements and fibres than initially expected, as even a relatively course mesh of elements and the cross-section yields suitably accurate results. The number of elements significantly affected the computational run time of the model, increasing with the number of elements in the model. The increase in fibre number also increased the run time, but to a lesser degree. The steel frame with 18 elements and 17 fibres for each cross-section sufficiently captured the structural behaviour of the frame. Note that these results do not specify the exact number of elements and fibres necessary for every scenario. The mesh required will be unique for every structural problem and it is therefore up to the discretion of the modeller to determine the optimum number of fibres when performing an analysis.

As cross-sections become more complicated with concrete cracking, temperature profiles, material yielding and other material behaviour it will necessitate the inclusion of more fibres. Also, the element number required for an analysis is greatly influenced by the temperature distribution. A structure with a more constant temperature distribution would require less elements than one with thermal gradients along the lengths. Similarly, a non-uniform temperature gradient along the cross-section height would require a finer fibre discretisation than a uniform temperature profile.

5.5 Conclusion

This chapter has shown how the FBE model built in OpenSees can successfully predict structural fire behaviour in various scenarios. Case Study 1 verified that the FBE model in OpenSees predicts deflections similar to those of a beam model of Vulcan presented in literature of a simple structure with non-uniform temperature gradient along the length of the beam as well as an applied axial force.

Case Study 2 was performed to test how the FBE model in OpenSees (FBE version 1) compares to the original FBE model (FBE version 0) and a more complex shell model in OpenSees. It also highlighted the application of the FBE model in composite structures with a non-linear temperature distribution along the height of the section. The predicted deflections of FBE version 1 were similar to FBE version 0 and the OpenSees shell model. However, all the models over-predicted the deflections when compared to the experimental results, although possible reasons for this are provided. A discretisation study performed on Test 15 of this case study revealed that a relatively coarse mesh of fibres in the cross-section would be sufficient for a FBE analysis.

Case Study 3 was the most complex benchmark that was modelled. It applied the FBE model in OpenSees to a global analysis of a steel frame with typical effects in fire: structural continuity, thermal bowing, axial forces and thermal gradients along the length and height of a beam. The predicted displacements and axial forces were compared to the more advanced software beam models of CEFICOSS, SAFIR and ABAQUS, as well as to experimental results, if available. The FBE model predicted trends in structural behaviour comparable to the advanced software models. Minor differences could be a result of variation in temperature inputs and material models adopted by the software. The axial force in the beam predicted by FBE showed closer results to the advanced models than a similar LS-DYNA beam element model.

A challenge for Case Study 3 is to accurately specify the thermal load. The FBE model does not perform a thermal analysis and requires exact temperature-time input. The temperatures in this case were taken from a SAFIR thermal analysis, as this represented a good average between the other models and provided a detailed temperature-time history. However, a temperature sensitivity study was performed on the FBE model and found that by decreasing or increasing the thermal load by 20% has a significant effect on the structural behaviour and predicted failure time. This suggests that more emphasis should be placed on the accuracy of temperature inputs rather than the complexity of the FBE model itself. However, the use of a simplified modelling technique, such as the FBE, allows sensitivity or statistical analyses to be carried out more readily.

A discretisation study was performed that separately investigated the effect of the number of elements and number of fibres on the predicted results. The study reveals that a relatively coarse mesh would yield results of sufficient accuracy, provided that the discretisation reflects the non-linearity of the thermal load. A comparison on the computational run time of the models reveals that a finer discretisation of fibres

would increase the computational effort of the analysis, although not in an exponential manner as per increasing the number of global DOFs. As the FBE formulation aims to be accurate, but still simple and computationally efficient, it is important to find an optimum number of elements and fibres for any FBE analysis.

Overall, the work provided in this chapter has extended the original FBE work by validating the FBE model in cases where there is axial restraint, member continuity and member buckling. Furthermore, the inclusion of studies into the discretisation of cross-sections and structures provides insight into the performance of the method relative to modelling techniques employed.

Chapter 6

Conclusions and recommendations

6.1 Overview

The aim of this research thesis was the development of the Fire Beam Element (FBE) method for the analysis of structures subjected to fire, by expanding it to be able to analyse global structures. This was done by implementing the FBE methodology into the finite element software OpenSees and validating it using three case studies. The FBE implementation in OpenSees was further used as a tool for investigating the effect of temperature inputs and mesh size on the analysis of structures in fire.

Chapter 1 highlights the need for the FBE method as an analysis tool for structures in fire. At this stage, the complexity of structural fire analysis is a wide spectrum ranging from very simplified basic methods to complicated advanced modelling. The FBE methodology aims to be a compromise between the two methods, by being a performance-based design approach that is simple and easy to use, while still capturing the true behaviour of structures in fire.

Chapter 2 explains how the FBE methodology is categorised under the broader field of fire safety and structural fire engineering. The chapter also provided background on fire and material models adopted in this research. Thereafter, the differences between prescriptive and performance-based design, ambient and fire design, coupled and decoupled analyses and local and global analyses are clarified. Important concepts in structural fire analysis are explained, such as the effect of structural continuity and fire testing. Lastly, the various software currently available for fire design are discussed.

Chapter 3 provides the theory and fundamental principles of the FBE formulation. This includes explaining the development of thermal strains and stresses and their conversion into equivalent thermal forces. The concept of a shifting neutral axis and the iterative procedure of a changing cross-sectional stiffness is outlined with the help of a simple example. Theses concepts are further integrated in the theory of finite element analysis (FEA) by using matrix manipulation. The chapter ends by discussing the benefits and limitations of the FBE method.

Building on the background and theory explained in Chapter 3, the FBE methodology was developed with the open-source FEA software OpenSees and Chapter 4 outlines how this was done. Firstly, OpenSees as a software is introduced and the reasons for using this platform are provided. The structure of the OpenSees framework is explained to assist readers in understanding the changes made in the code of OpenSees to accommodate the FBE methodology. The code was written in C++ in Microsoft Visual Studio. The new classes that were developed in this research, called *FireFiberSection* and *FireEl*, are explained and their functions described.

Chapter 5 aims to validate the FBE method for the analysis of structures in fire by considering three case studies in order of increasing complexity. The last case study was used to perform studies to analyse the temperature sensitivity on predicted deflections as well as the effect of mesh size on the computational run time of the analysis. The conclusions drawn from these case studies are further discussed in Section 6.3.

Lastly, Chapter 6 provides a brief summary of the previous chapters. The main findings from Chapter 5 are discussed and the recommendations for future research for the FBE concept are made.

6.2 Consideration of objectives

The main objective of this thesis was the development of the FBE methodology for global modelling of structures in fire. This was satisfied by adressing the aims, as previously listed in Section 1.3, as follows:

- The theory of the FBE formulation and the derivation of the matrix manipulation accounting for the updated neutral axis (NA) position is provided in Chapter 3. The integration of the basic FBE principles into FEA theory is shown, allowing for the FBE methodology to be applied to statically indeterminate structures with restraint.
- The FBE formulation was implemented into OpenSees, an open-source FEA software.
- Simple, user-friendly commands for building a FBE skeletal frame in OpenSees were developed.
- Three case studies were performed for the validation of the FBE model.
- Parametric studies were conducted to investigate the effect of temperature on predicted deflections and the effect of the mesh size on the computational run time of the FBE analysis of a skeletal frame.

6.3 Summary of findings

6.3.1 Implementation of FBE into OpenSees

The focus of this work was the development of the FBE formulation into the open-source FEA software OpenSees. This was done by (a) modifying functions and creating new subclasses of the existing base classes in the C++ code in the development platform of OpenSees, and (b) specifying commands in Tool Command Language (TCL) that can be called by an OpenSees user.

The two main subclasses *FireFiberSection* and *FireEl* were developed in OpenSees with the aim of creating an intelligent beam element that updates the position of its NA. The *FireFiberSection* is a subclass of the base class *SectionForceDeformation*, which embodies the functions and properties of the cross-section of an element. The section is defined by multiple fibres and each fibre can be assigned a different material and temperature. *FireFiberSection* contains a function *calculateC*, which determines the position of the NA relative to the reference axis (RA) after every iteration. As another contribution, the existing code was modified to allow the input of 25 temperature points along the height of a cross-section, as opposed to 2, 5 or 9 in the existing code. This was developed with the aim of modelling the non-linear temperature profile of a cross-section (especially of a composite section) more accurately.

FireEl was developed as a subclass of the base class *Element* in the OpenSees code. This element is assigned a cross-section *FireFiberSection* and can therefore determine the update position of the NA after every iteration. The functions *getTangentStiff* and *addLoad* apply the eccentricity of the NA (c') and use matrix manipulation to update the stiffness matrix and load vector respectively.

An OpenSees user builds a model in OpenSees by writing a TCL script which links to the code written in C++. Therefore, special TCL commands were created to enable the user to build a FBE skeletal model in OpenSees. These commands were written to call the classes *FireFiberSection* and *FireEl*. As a result, this work provides a user-friendly option of analysing structures in fire using FBE skeletal frames without the need to code in C++ or access the source code.

6.3.2 Validation study

Three case studies in order of increasing complexity were chosen as a validation study of the FBE model in OpenSees software. The last case study extended the original FBE work by including high axial loads, structural continuity and buckling.

Case study 1 was based on a COST (2014) benchmark of a simply-supported beam with non-linear heating along its length as well as an applied vertical and axial load. The deflections predicted by the FBE model show a close correlation to the Vulcan model, differing by only 4% on average.

Case Study 2 was chosen to validate the FBE model in OpenSees to the original FBE model (verion 0) in Prokon and the OpenSees shell model. The case study modelled two fire tests of a simply-supported composite beam, known as Test 15 and Test 16, from Wainman and Kirby (1989). The deflections predicted by the FBE model are almost identical to the original version, and very similar to the OpenSees shell model. This demonstrates that the FBE model can successfully analyse composite structures subjected to non-linear heating along the height of a cross-section. However, when comparing the predictions to the experimental deflections, it can be seen that all models over-predicted the deflections. Possible reasons for this are provided.

The last case study was a 2D global analysis of a steel frame based on one of the famous fire tests performed in Cardington (Franssen *et al.*, 1995). The structure was subjected to a number of complex effects caused by a fire: thermal gradients along the length and cross-sectional height of a element leading to thermal bowing, restraint induced by a cooler adjacent structure and buckling of the beam. The FBE model is compared to advanced software models CEFICOSS, SAFIR and ABAQUS, as well as to a LS-DYNA beam model and the experimental results. The predictions of the deflections and axial forces by the FBE model show good agreement to the advanced models and the experiment. Interestingly, the FBE model is able to predict the axial force and buckling of the beam to a greater accuracy than the LS-DYNA beam element. For example, at a certain point in time the FBE model differed by 4% to the other models, while the LS-DYNA model differed by more than 23% at the same point.

6.3.3 Parametric study

The advantage of the implementation of FBE in OpenSees is that the model is simple and requires little computational effort. This allows for sensitivity and parametric studies to be performed relatively quickly and modellers can easily investigate the effect of changing certain parameters. This is especially important in structural fire analysis, where there is great uncertainty in the fire load, as explained in Section 2.6.1. Examples of such a study on Case Study 3 were presented in this work.

A temperature sensitivity study on the steel frame from Case Study 3 revealed that the behaviour of the structure is significantly affected by the temperature. This suggests that various temperature profiles should be modelled for a structural fire analysis, and the simplicity of the FBE model makes this feasible.

A discretisation study on the steel frame investigated the effect of the number of fibres and the number of elements on the predicted results and analysis run times. A relatively coarse mesh was sufficient to accurately predict the structural response of the steel frame. It was found that a higher number of elements significantly increased the computational run-time of the analysis, which is undesirable, especially in large structures. However, such a phenomena occurs in all models with increasing degrees of freedom (DOFs), and it not specific to this research. Increasing the number of fibres in a cross-section showed only minor increases in computational run time. The increase in computational time required to analyse a cross-section is approximately

proportional to the number of fibres (i.e. DOFs), rather than being proportional to the square of the number of DOFs as per global analyses. The use of a finer fibre mesh would be more desirable as non-linear thermal gradients along the height of a cross-section can be specified accurately. Therefore, a coarser mesh of elements and a finer mesh of fibres for the cross-sections is recommended by this research, as this would be efficient and accurate for the analysis of large structures in fire. However, this will be highly dependent on the nature of the structure analysed.

6.4 Future work

The research of the FBE methodology has been extended from the analysis of simply-supported beams in the original work of Walls (2016) to the global analysis of 2D structures with more complex effects of restraint, axial loads and buckling. The next step would be the development of a 3D FBE model in OpenSees. This would require the consideration of a 3D stiffness matrix and load vector of a beam in relationship to the reference axis (RA) (similar to the derivation in Section 3.6.2, but with additional DOFs). The matrix operator applied to the stiffness matrix and load vector to account for the shift in NA would have to be mathematically derived. Thereafter, a 3D version of the class *FireEl* could be implemented in OpenSees. With sufficient case studies and validation, this model could potentially be used in industry for the analysis and design of 3D structures subjected to fire.

As suggested by Walls *et al.* (2017), the FBE methodolgy could be applied in conjuction with the Slab Panel Method (SPM) developed by Clifton and Abu (2014). In the fire analysis of a 3D frame, the FBE model would perform the analysis of the skeletal frame, as displayed in Figure 3.15, while the SPM conducts the analysis of the slabs. Further research and development in the integration of these two methodologies for structures in fire is recommended. Following the research of this thesis, this could possibly be built into OpenSees.

In addition to the above, future research should focus on the following areas:

- Investigating the inclusion of slippage in composite members in a fire.
- Consideration of lateral buckling behaviour by including design codes or sub-functions to analyse cross-sectional buckling.
- Inclusion of joint behaviour in the FBE model.
- Performing statistical analyses of structures using the FBE method.
- Implementation of various load scenarios such as thermal gradients in OpenSees.

6.5 Closing comments

The work presented in this thesis contributed to the development towards using beam elements for the global modelling of structures subjected to fire. With further research and development in this field, especially applying the FBE formulation

to 3D models in conjunction with the SPM method, the FBE formulation shows potential as a feasible performance-based modelling tool for structural fire analyses in industry. This would potentially minimise the costly use of passive protection (such as intumescent paints and boards) against fire, whilst reducing the modelling time of a fire analysis. By easily changing certain parameters such as temperatures, designers can run sensitivity analyses relatively quickly. They can thereby gain an understanding of the behaviour of a structure in fire and design according to the worst-case scenario. Considering these advantages, the FBE methodology could contribute to the overall aim of fire safety in the design of safer and more economical buildings.

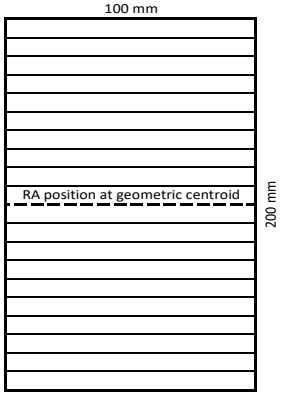
Appendices

Appendix A

Calculations of Example A

This appendix contains details of Example A, a simple cantilever subjected to a linear temperature profile, as discussed in Section 3.5.3. Table A.1 shows how the cross-section was discretised into 20 fibres. This is followed by Tables A.2 to A.4 which are the iterative calculations performed in Microsoft Excel for cross-sectional convergence.

Table A.1: Geometric and material properties of each fibre in the cross-section of Example A.



	Geometric Properties					Material Properties				
Fibre number i	w (mm)	t (mm)	A_i (mm^2)	Position of centroid from RA (mm)	I_i around fibre centroid (mm^4)	$f_{y,20}$ (MPa)	E_{20} (MPa)	T (°C)	$f_{y,\theta}$ (MPa)	E_θ (MPa)
1	100	10	1000	95	8333	355	200000	43.2	355.0	200000
2	100	10	1000	85	8333	355	200000	89.7	355.0	200000
3	100	10	1000	75	8333	355	200000	136.3	355.0	192750
4	100	10	1000	65	8333	355	200000	182.8	355.0	183450
5	100	10	1000	55	8333	355	200000	229.3	355.0	174150
6	100	10	1000	45	8333	355	200000	275.8	355.0	164850
7	100	10	1000	35	8333	355	200000	322.3	355.0	155550
8	100	10	1000	25	8333	355	200000	368.8	355.0	146250
9	100	10	1000	15	8333	355	200000	415.3	343.1	136950
10	100	10	1000	5	8333	355	200000	461.8	306.8	127650
11	100	10	1000	-5	8333	355	200000	508.3	267.8	115215
12	100	10	1000	-15	8333	355	200000	554.8	216.6	88245
13	100	10	1000	-25	8333	355	200000	601.3	165.8	61550
14	100	10	1000	-35	8333	355	200000	647.8	126.2	44810
15	100	10	1000	-45	8333	355	200000	694.3	86.5	28070
16	100	10	1000	-55	8333	355	200000	740.8	64.3	22740
17	100	10	1000	-65	8333	355	200000	787.3	44.5	19020
18	100	10	1000	-75	8333	355	200000	833.8	33.1	16481
19	100	10	1000	-85	8333	355	200000	880.3	24.8	14389
20	100	10	1000	-95	8333	355	200000	926.8	19.4	12296
Total section	100	200	20000	0	66666667	355	200000			

Table A.2: Calculations of the initial iteration for Example A

	Iteration 0, j = 0																	
Fibre number i	y_i	ε_θ	σ_θ	M_θ	N_θ	E_θ	EA_θ	I_i around NA	EI_θ	ε_T	ε_σ	E_s	σ	EA	EAy	c'
	mm	(mm/mm)	(MPa)	(Nmm)	(N)	(MPa)	(MPa mm^2)	(mm^4)	(MPa mm^4)	(mm/mm)	(mm/mm)	(MPa)	(MPa)	(MPa mm^2)	(MPa mm^3)	mm
1	95	0.0003	56.98	5.41E+06	56976	200000	2.00E+08	9.03E+06	1.81E+12	0.0047	0.0044	81165	355.00	8.12E+07	7.71E+09	39.3
2	85	0.0009	173.52	1.47E+07	173524	200000	2.00E+08	7.23E+06	1.45E+12	0.0046	0.0037	95398	355.00	9.54E+07	8.11E+09	
3	75	0.0015	282.89	2.12E+07	282891	192750	1.93E+08	5.63E+06	1.09E+12	0.0045	0.0031	111221	327.08	1.11E+08	8.34E+09	
4	65	0.0021	382.49	2.49E+07	382491	183450	1.83E+08	4.23E+06	7.77E+11	0.0044	0.0024	132613	287.59	1.33E+08	8.62E+09	
5	55	0.0027	473.62	2.60E+07	473622	174150	1.74E+08	3.03E+06	5.28E+11	0.0044	0.0017	165611	239.35	1.66E+08	9.11E+09	
6	45	0.0034	555.80	2.50E+07	555800	164850	1.65E+08	2.03E+06	3.35E+11	0.0043	0.0009	164850	127.46	1.65E+08	7.42E+09	
7	35	0.0040	628.54	2.20E+07	628543	155550	1.56E+08	1.23E+06	1.92E+11	0.0042	0.0002	155550	24.07	1.56E+08	5.44E+09	
8	25	0.0047	691.37	1.73E+07	691369	146250	1.46E+08	6.33E+05	9.26E+10	0.0042	-0.0006	146250	-59.61	1.46E+08	3.66E+09	
9	15	0.0054	743.79	1.12E+07	743793	136950	1.37E+08	2.33E+05	3.20E+10	0.0041	-0.0013	126274	-115.09	1.26E+08	1.89E+09	
10	5	0.0062	785.33	3.93E+06	785335	127650	1.28E+08	3.33E+04	4.26E+09	0.0040	-0.0021	86818	-117.58	8.68E+07	4.34E+08	
11	-5	0.0069	793.91	-3.97E+06	793909	115215	1.15E+08	3.33E+04	3.84E+09	0.0040	-0.0029	61376	-102.00	6.14E+07	-3.07E+08	
12	-15	0.0076	674.76	-1.01E+07	674756	88245	8.82E+07	2.33E+05	2.06E+10	0.0039	-0.0038	40459	-65.78	4.05E+07	-6.07E+08	
13	-25	0.0084	518.21	-1.30E+07	518214	61550	6.16E+07	6.33E+05	3.90E+10	0.0038	-0.0046	25931	-36.18	2.59E+07	-6.48E+08	
14	-35	0.0092	412.69	-1.44E+07	412688	44810	4.48E+07	1.23E+06	5.53E+10	0.0038	-0.0055	17337	-20.96	1.73E+07	-6.07E+08	
15	-45	0.0100	281.19	-1.27E+07	281187	28070	2.81E+07	2.03E+06	5.71E+10	0.0037	-0.0063	10538	-15.11	1.05E+07	-4.74E+08	
16	-55	0.0108	246.55	-1.36E+07	246553	22740	2.27E+07	3.03E+06	6.90E+10	0.0036	-0.0072	7254	-9.27	7.25E+06	-3.99E+08	
17	-65	0.0110	209.22	-1.36E+07	209220	19020	1.90E+07	4.23E+06	8.05E+10	0.0035	-0.0075	5047	-3.43	5.05E+06	-3.28E+08	
18	-75	0.0110	181.29	-1.36E+07	181294	16481	1.65E+07	5.63E+06	9.28E+10	0.0035	-0.0075	3813	-2.15	3.81E+06	-2.86E+08	
19	-85	0.0114	164.10	-1.39E+07	164104	14389	1.44E+07	7.23E+06	1.04E+11	0.0034	-0.0080	2790	-1.36	2.79E+06	-2.37E+08	
20	-95	0.0123	151.67	-1.44E+07	151674	12296	1.23E+07	9.03E+06	1.11E+11	0.0033	-0.0090	2001	-0.89	2.00E+06	-1.90E+08	
Total section				48409813	8407943		2.10E+09	6.67E+07	6.93E+12					1.44E+09	5.67E+10	

Table A.3: Calculations of the first iteration for Example A

Iteration 1, j = 1																
Fibre number i	y_i	ε_θ	σ_θ	M_θ	N_θ	E_s^{j-1}	EA_θ	I_i around NA	EI_θ	ε_T	ε_σ	E_s	σ	EA	EAy	c'
	mm	(mm/mm)	(MPa)	(Nmm)	(N)	(MPa)	(MPa mm^2)	(mm^4)	(MPa mm^4)	(mm/mm)	(mm/mm)	(MPa)	(MPa)	(MPa mm^2)	(MPa mm^3)	mm
1	55.7	0.0003	23.12	1.29E+06	23122	81165	8.12E+07	3.11E+06	2.53E+11	0.0001	-0.0002	200000	-35.29	2.00E+08	1.90E+10	37.4
2	45.7	0.0009	82.77	3.78E+06	82769	95398	9.54E+07	2.10E+06	2.00E+11	0.0008	-0.0001	200000	-17.86	2.00E+08	1.70E+10	
3	35.7	0.0015	163.23	5.83E+06	163235	111221	1.11E+08	1.28E+06	1.43E+11	0.0014	0.0000	192750	3.60	1.93E+08	1.45E+10	
4	25.7	0.0021	276.50	7.11E+06	276498	132613	1.33E+08	6.70E+05	8.88E+10	0.0021	0.0000	183450	5.58	1.83E+08	1.19E+10	
5	15.7	0.0027	450.40	7.08E+06	450398	165611	1.66E+08	2.55E+05	4.23E+10	0.0028	0.0001	174150	10.38	1.74E+08	9.58E+09	
6	5.7	0.0034	555.80	3.18E+06	555800	164850	1.65E+08	4.11E+04	6.77E+09	0.0035	0.0001	164850	11.74	1.65E+08	7.42E+09	
7	-4.3	0.0040	628.54	-2.69E+06	628543	155550	1.56E+08	2.66E+04	4.15E+09	0.0041	0.0001	155550	10.54	1.56E+08	5.44E+09	
8	-14.3	0.0047	691.37	-9.87E+06	691369	146250	1.46E+08	2.12E+05	3.10E+10	0.0048	0.0001	146250	7.54	1.46E+08	3.66E+09	
9	-24.3	0.0054	685.81	-1.67E+07	685812	126274	1.26E+08	5.98E+05	7.55E+10	0.0055	0.0000	136950	3.43	1.37E+08	2.05E+09	
10	-34.3	0.0062	534.13	-1.83E+07	534125	86818	8.68E+07	1.18E+06	1.03E+11	0.0061	0.0000	127650	1.19	1.28E+08	6.38E+08	
11	-44.3	0.0069	422.92	-1.87E+07	422924	61376	6.14E+07	1.97E+06	1.21E+11	0.0068	-0.0001	115215	-5.52	1.15E+08	-5.76E+08	
12	-54.3	0.0076	309.37	-1.68E+07	309369	40459	4.05E+07	2.95E+06	1.20E+11	0.0075	-0.0002	88245	-6.58	8.82E+07	-1.32E+09	
13	-64.3	0.0084	218.32	-1.40E+07	218320	25931	2.59E+07	4.14E+06	1.07E+11	0.0081	-0.0003	61550	-5.15	6.16E+07	-1.54E+09	
14	-74.3	0.0092	159.67	-1.19E+07	159672	17337	1.73E+07	5.53E+06	9.58E+10	0.0088	-0.0004	44810	-3.94	4.48E+07	-1.57E+09	
15	-84.3	0.0100	105.57	-8.90E+06	105567	10538	1.05E+07	7.11E+06	7.49E+10	0.0095	-0.0005	28070	-2.09	2.81E+07	-1.26E+09	
16	-94.3	0.0108	78.65	-7.41E+06	78647	7254	7.25E+06	8.90E+06	6.45E+10	0.0102	-0.0007	22740	-1.77	2.27E+07	-1.25E+09	
17	-104.3	0.0110	55.51	-5.79E+06	55512	5047	5.05E+06	1.09E+07	5.49E+10	0.0108	-0.0002	19020	-0.31	1.90E+07	-1.24E+09	
18	-114.3	0.0110	41.95	-4.79E+06	41946	3813	3.81E+06	1.31E+07	4.98E+10	0.0115	0.0005	16481	0.67	1.65E+07	-1.24E+09	
19	-124.3	0.0114	31.82	-3.95E+06	31822	2790	2.79E+06	1.55E+07	4.31E+10	0.0122	0.0008	14389	0.79	1.44E+07	-1.22E+09	
20	-134.3	0.0123	24.69	-3.31E+06	24686	2001	2.00E+06	1.80E+07	3.61E+10	0.0128	0.0005	12296	0.38	1.23E+07	-1.17E+09	
Total section				-1.15E+08	5540137		1.44E+09	9.75E+07	1.7141E+12					2.10E+09	7.88E+10	

Table A.4: Calculations of the final iteration for Example A

	Iteration 2, j = 2																	
Fibre number i	y_i	ε_θ	σ_θ	M_θ	N_θ	E_s^{j-1}	EA_θ	I_i around NA	EI_θ	ε_T	ε_σ	E_s	σ	EA	EAy	c'
	mm	(mm/mm)	(MPa)	(Nmm)	(N)	(MPa)	(MPa mm^2)	(mm^4)	(MPa mm^4)	(mm/mm)	(mm/mm)	(MPa)	(MPa)	(MPa mm^2)	(MPa mm^3)	mm
1	57.6	0.0003	56.98	3.28E+06	56976	200000	2.00E+08	3.32E+06	6.64E+11	0.0001	-0.0001	200000	-27.70	2.00E+08	1.90E+10	37.4
2	47.6	0.0009	173.52	8.25E+06	173524	200000	2.00E+08	2.27E+06	4.54E+11	0.0008	-0.0001	200000	-10.51	2.00E+08	1.70E+10	
3	37.6	0.0015	282.89	1.06E+07	282891	192750	1.93E+08	1.42E+06	2.74E+11	0.0015	0.0000	192750	2.98	1.93E+08	1.45E+10	
4	27.6	0.0021	382.49	1.05E+07	382491	183450	1.83E+08	7.68E+05	1.41E+11	0.0022	0.0001	183450	11.34	1.83E+08	1.19E+10	
5	17.6	0.0027	473.62	8.32E+06	473622	174150	1.74E+08	3.17E+05	5.52E+10	0.0028	0.0001	174150	15.38	1.74E+08	9.58E+09	
6	7.6	0.0034	555.80	4.20E+06	555800	164850	1.65E+08	6.55E+04	1.08E+10	0.0035	0.0001	164850	16.06	1.65E+08	7.42E+09	
7	-2.4	0.0040	628.54	-1.53E+06	628543	155550	1.56E+08	1.43E+04	2.22E+09	0.0042	0.0001	155550	14.23	1.56E+08	5.44E+09	
8	-12.4	0.0047	691.37	-8.60E+06	691369	146250	1.46E+08	1.63E+05	2.38E+10	0.0048	0.0001	146250	10.67	1.46E+08	3.66E+09	
9	-22.4	0.0054	743.79	-1.67E+07	743793	136950	1.37E+08	5.12E+05	7.01E+10	0.0055	0.0001	136950	6.06	1.37E+08	2.05E+09	
10	-32.4	0.0062	785.33	-2.55E+07	785335	127650	1.28E+08	1.06E+06	1.35E+11	0.0062	0.0000	127650	0.99	1.28E+08	6.38E+08	
11	-42.4	0.0069	793.91	-3.37E+07	793909	115215	1.15E+08	1.81E+06	2.08E+11	0.0068	-0.0001	115215	-3.82	1.15E+08	-5.76E+08	
12	-52.4	0.0076	674.76	-3.54E+07	674756	88245	8.82E+07	2.76E+06	2.43E+11	0.0075	-0.0001	88245	-5.63	8.82E+07	-1.32E+09	
13	-62.4	0.0084	518.21	-3.24E+07	518214	61550	6.16E+07	3.91E+06	2.40E+11	0.0082	-0.0002	61550	-4.72	6.16E+07	-1.54E+09	
14	-72.4	0.0092	412.69	-2.99E+07	412688	44810	4.48E+07	5.26E+06	2.36E+11	0.0088	-0.0004	44810	-3.72	4.48E+07	-1.57E+09	
15	-82.4	0.0100	281.19	-2.32E+07	281187	28070	2.81E+07	6.80E+06	1.91E+11	0.0095	-0.0005	28070	-2.01	2.81E+07	-1.26E+09	
16	-92.4	0.0108	246.55	-2.28E+07	246553	22740	2.27E+07	8.55E+06	1.94E+11	0.0102	-0.0007	22740	-1.72	2.27E+07	-1.25E+09	
17	-102.4	0.0110	209.22	-2.14E+07	209220	19020	1.90E+07	1.05E+07	2.00E+11	0.0108	-0.0002	19020	-0.28	1.90E+07	-1.24E+09	
18	-112.4	0.0110	181.29	-2.04E+07	181294	16481	1.65E+07	1.27E+07	2.08E+11	0.0115	0.0005	16481	0.70	1.65E+07	-1.24E+09	
19	-122.4	0.0114	164.10	-2.01E+07	164104	14389	1.44E+07	1.50E+07	2.16E+11	0.0122	0.0008	14389	0.80	1.44E+07	-1.22E+09	
20	-132.4	0.0123	151.67	-2.01E+07	151674	12296	1.23E+07	1.75E+07	2.16E+11	0.0129	0.0005	12296	0.39	1.23E+07	-1.17E+09	
Total section				-2.66E+08	8407943		1.44E+09	9.75E+07	3.98E+12					2.10E+09	7.88E+10	

Appendix B

Modelling in OpenSees: An Example

This appendix contains an example displaying the input and analysis components of an analysis performed in OpenSees. The example is a simply-supported beam subjected to a horizontal and vertical point load as well as a thermal load. This case study was investigated in Chapter 5 as Case Study 1 in Section 5.2.

B.1 Input

The following page is the TCL input script for the analysis of Case Study 1 from Section 5.2.

```
# Example script of a simply supported steel beam in a furnace with vertical and horizontal load
# Model and Nodes
model BasicBuilder -ndm 2 - ndf 3
node    1       0       0
node    2       0.25    0
node    3       0.575   0
node    4       0.9     0
node    5       1.2     0
node    6       1.5     0
node    7       1.8     0
node    8       2.1     0
node    9       2.425   0
node    10      2.75    0
node    11      3       0
# Boundary conditions
fix     2       1       1       0
fix     10      0       1       0
# Material and Sections
uniaxialMaterial SteelECThermal 1 EC3 362.4e6 209e9; #Steel according to Eurocode 3
section FireFiberSection 1 {
fiber   0.102   0       0.00144         1
fiber   0.094   0       0.00144         1
fiber   0.08    0       0.00032         1
fiber   0.06    0       0.00032         1
fiber   0.04    0       0.00032         1
fiber   0.02    0       0.00032         1
fiber   0       0       0.00032         1
fiber   -0.02   0       0.00032         1
fiber   -0.04   0       0.00032         1
fiber   -0.06   0       0.00032         1
fiber   -0.08   0       0.00032         1
fiber   -0.094  0       0.00144         1
fiber   -0.102  0       0.00144         1 }
```

```
#Elements
geomTransf Corotational 1
for {set i 1} {$i<11} {incr i} {
        element FireEl $i $i [ expr $i+1] 3 1 1
}
#Loads
set V -200000; #vertical load
set H -400000; #horizontal load
#applied  mechanical loads
pattern Plain 1 Constant {
load 6 0 $V 0
load 11 $H 0 0
}
#furnace load
set     Y25     0.105
set     Y24     0.1
set     Y23     0.096
set     Y22     0.092
set     Y21     0.084705882
set     Y20     0.074117647
set     Y19     0.063529412
set     Y18     0.052941176
set     Y17     0.042352941
set     Y16     0.031764706
set     Y15     0.021176471
set     Y14     0.010588235
set     Y13     0
set     Y12     -0.010588235
set     Y11     -0.021176471
set     Y10     -0.031764706
set     Y9      -0.042352941
set     Y8      -0.052941176
```

```
set     Y7      -0.063529412

set     Y6      -0.074117647

set     Y5      -0.084705882

set     Y4      -0.092

set     Y3      -0.096

set     Y2      -0.1

set     Y1      -0.105

pattern Plain 2 Linear {

eleLoad -ele 1 10 -type -beamThermal -source BMS3_T1.dat $Y1 $Y2 $Y3 $Y4 $Y5 $Y6 $Y7 $Y8 $Y9
$Y10 $Y11 $Y12 $Y13 $Y14 $Y15 $Y16 $Y17 $Y18 $Y19 $Y20 $Y21 $Y22 $Y23 $Y24 $Y25

eleLoad -ele 2 9 -type -beamThermal -source BMS3_T2.dat $Y1 $Y2 $Y3 $Y4 $Y5 $Y6 $Y7 $Y8 $Y9
$Y10 $Y11 $Y12 $Y13 $Y14 $Y15 $Y16 $Y17 $Y18 $Y19 $Y20 $Y21 $Y22 $Y23 $Y24 $Y25

eleLoad -ele 3 8 -type -beamThermal -source BMS3_T3.dat $Y1 $Y2 $Y3 $Y4 $Y5 $Y6 $Y7 $Y8 $Y9
$Y10 $Y11 $Y12 $Y13 $Y14 $Y15 $Y16 $Y17 $Y18 $Y19 $Y20 $Y21 $Y22 $Y23 $Y24 $Y25

eleLoad -ele 4 -type -beamThermal -source BMS3_T4.dat $Y1 $Y2 $Y3 $Y4 $Y5 $Y6 $Y7 $Y8 $Y9 $Y10
$Y11 $Y12 $Y13 $Y14 $Y15 $Y16 $Y17 $Y18 $Y19 $Y20 $Y21 $Y22 $Y23 $Y24 $Y25

eleLoad -ele 5 -type -beamThermal -source BMS3_T5.dat $Y1 $Y2 $Y3 $Y4 $Y5 $Y6 $Y7 $Y8 $Y9 $Y10
$Y11 $Y12 $Y13 $Y14 $Y15 $Y16 $Y17 $Y18 $Y19 $Y20 $Y21 $Y22 $Y23 $Y24 $Y25

eleLoad -ele 6 -type -beamThermal -source BMS3_T6.dat $Y1 $Y2 $Y3 $Y4 $Y5 $Y6 $Y7 $Y8 $Y9 $Y10
$Y11 $Y12 $Y13 $Y14 $Y15 $Y16 $Y17 $Y18 $Y19 $Y20 $Y21 $Y22 $Y23 $Y24 $Y25

eleLoad -ele 7 -type -beamThermal -source BMS3_T7.dat $Y1 $Y2 $Y3 $Y4 $Y5 $Y6 $Y7 $Y8 $Y9 $Y10
$Y11 $Y12 $Y13 $Y14 $Y15 $Y16 $Y17 $Y18 $Y19 $Y20 $Y21 $Y22 $Y23 $Y24 $Y25

}

#Output command

recorder Node -fileCSV COSTdeflection.csv -time -node 6 -dof 2 disp

#Analysis parameters :

system BandGeneral;

constraints Transformation;

numberer RCM;

test NormDispIncr 1e-4 100;

algorithm Newton;

integrator LoadControl 0.01;

analysis Static;

analyze 100
```

B.2 Analysis

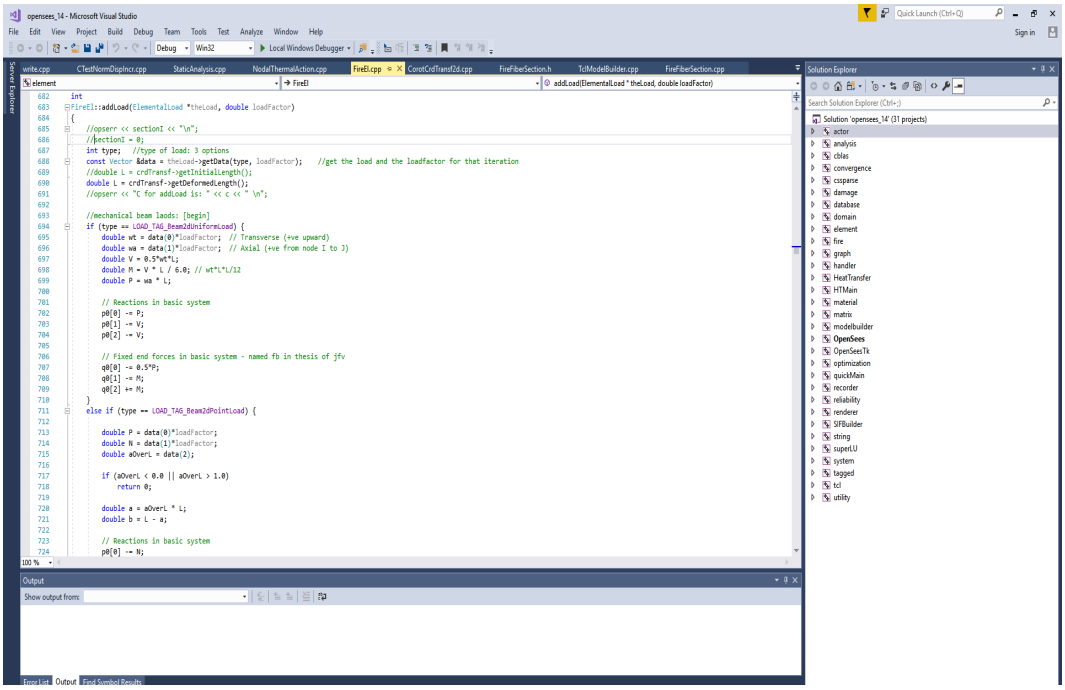


Figure B.1: Screen-shot of C++ code in Microsoft Visual Studio

Appendix C

Code written in C++ for OpenSees

This appendix contains the most important code written in C++ in Micrososft Visual Studio 2017. The comments are shown in green.

C.1 Code of functions in the class *FireFiberSection*

C.1.1 Function *determineFiberTemperature*

The code in C++ on the following page represents the function *determineFiberTemperature* which was modified to account for 25 temperature inputs. The fibre temperature is determined by linear interpolation between the 25 input points.

```
const Vector&
FireFiberSection::determineFiberTemperature(const Vector& DataMixed, double fiberLoc)
{
	double FiberTemperature = 0;
	double FiberTempMax = 0;
	double dataTempe[50];
	for (int i = 0; i < 50; i++) {
		dataTempe[i] = DataMixed(i);
		}
	if (fiberLoc <= dataTempe[1]){
		opserr << "FireFiberSection::determineTemp -- fiber loc is out of the
		section at the top end \n";
		opserr << dataTempe[1] << "\n";
		}
	else if (fiberLoc <= dataTempe[3]) {
		FiberTemperature = dataTempe[0] - (dataTempe[1] - fiberLoc)*
		(dataTempe[0] - dataTempe[2]) / (dataTempe[1] - dataTempe[3]);
		}
	else if (fiberLoc <= dataTempe[5]) {
		FiberTemperature = dataTempe[2] - (dataTempe[3] - fiberLoc) *
		(dataTempe[2] - dataTempe[4]) / (dataTempe[3] - dataTempe[5]);
		}
	else if (fiberLoc <= dataTempe[7]) {
		FiberTemperature = dataTempe[4] - (dataTempe[5] - fiberLoc) *
		(dataTempe[4] - dataTempe[6]) / (dataTempe[5] - dataTempe[7]);
		}
	else if (fiberLoc <= dataTempe[9]) {
		FiberTemperature = dataTempe[6] - (dataTempe[7] - fiberLoc) *
		(dataTempe[6] - dataTempe[8]) / (dataTempe[7] - dataTempe[9]);
		}
	else if (fiberLoc <= dataTempe[11]){
		FiberTemperature = dataTempe[8] - (dataTempe[9] - fiberLoc) *
		(dataTempe[8] - dataTempe[10]) / (dataTempe[9] - dataTempe[11]);
		}
	else if (fiberLoc <= dataTempe[13]){
		FiberTemperature = dataTempe[10] - (dataTempe[11] - fiberLoc) *
		(dataTempe[10] - dataTempe[12]) / (dataTempe[11] - dataTempe[13]);
		}
	else if (fiberLoc <= dataTempe[15]){
		FiberTemperature = dataTempe[12] - (dataTempe[13] - fiberLoc) *
		(dataTempe[12] - dataTempe[14]) / (dataTempe[13] - dataTempe[15]);
		}
	else if (fiberLoc <= dataTempe[17]){
		FiberTemperature = dataTempe[14] - (dataTempe[15] - fiberLoc) *
		(dataTempe[14] - dataTempe[16]) / (dataTempe[15] - dataTempe[17]);
		}
	else if (fiberLoc <= dataTempe[19]){
		FiberTemperature = dataTempe[16] - (dataTempe[17] - fiberLoc) * (
		dataTempe[16] - dataTempe[18]) / (dataTempe[17] - dataTempe[19]);
		}
	else if (fiberLoc <= dataTempe[21]){
		FiberTemperature = dataTempe[18] - (dataTempe[19] - fiberLoc) *
		(dataTempe[18] - dataTempe[20]) / (dataTempe[19] - dataTempe[21]);
		}
	else if (fiberLoc <= dataTempe[23]){
		FiberTemperature = dataTempe[20] - (dataTempe[21] - fiberLoc) *
		(dataTempe[20] - dataTempe[22]) / (dataTempe[21] - dataTempe[23]);
		}
	else if (fiberLoc <= dataTempe[25]){
		FiberTemperature = dataTempe[22] - (dataTempe[23] - fiberLoc) *
		(dataTempe[22] - dataTempe[24]) / (dataTempe[23] - dataTempe[25]);
		}
	else if (fiberLoc <= dataTempe[27]){
```

```
        FiberTemperature = dataTempe[24] - (dataTempe[25] - fiberLoc) *
        (dataTempe[24] - dataTempe[26]) / (dataTempe[25] - dataTempe[27]);
        }
    else if (fiberLoc <= dataTempe[29]){
        FiberTemperature = dataTempe[26] - (dataTempe[27] - fiberLoc) *
        (dataTempe[26] - dataTempe[28]) / (dataTempe[27] - dataTempe[29]);
        }
    else if (fiberLoc <= dataTempe[31]){
        FiberTemperature = dataTempe[28] - (dataTempe[29] - fiberLoc) *
        (dataTempe[28] - dataTempe[30]) / (dataTempe[29] - dataTempe[31]);
        }
    else if (fiberLoc <= dataTempe[33]){
        FiberTemperature = dataTempe[30] - (dataTempe[31] - fiberLoc) *
        (dataTempe[30] - dataTempe[32]) / (dataTempe[31] - dataTempe[33]);
        }
    else if (fiberLoc <= dataTempe[35]){
        FiberTemperature = dataTempe[32] - (dataTempe[33] - fiberLoc) *
        (dataTempe[32] - dataTempe[34]) / (dataTempe[33] - dataTempe[35]);
        }
    else if (fiberLoc <= dataTempe[37]){
        FiberTemperature = dataTempe[34] - (dataTempe[35] - fiberLoc) *
        (dataTempe[34] - dataTempe[36]) / (dataTempe[35] - dataTempe[37]);
        }
    else if (fiberLoc <= dataTempe[39]){
        FiberTemperature = dataTempe[36] - (dataTempe[37] - fiberLoc) *
        (dataTempe[36] - dataTempe[38]) / (dataTempe[37] - dataTempe[39]);
        }
    else if (fiberLoc <= dataTempe[41]){
        FiberTemperature = dataTempe[38] - (dataTempe[39] - fiberLoc) *
        (dataTempe[38] - dataTempe[40]) / (dataTempe[39] - dataTempe[41]);
        }
    else if (fiberLoc <= dataTempe[43]){
        FiberTemperature = dataTempe[40] - (dataTempe[41] - fiberLoc) *
        (dataTempe[40] - dataTempe[42]) / (dataTempe[41] - dataTempe[43]);
        }
    else if (fiberLoc <= dataTempe[45]){
        FiberTemperature = dataTempe[42] - (dataTempe[43] - fiberLoc) *
        (dataTempe[42] - dataTempe[44]) / (dataTempe[43] - dataTempe[45]);
        }
    else if (fiberLoc <= dataTempe[47]){
        FiberTemperature = dataTempe[44] - (dataTempe[45] - fiberLoc) *
        (dataTempe[44] - dataTempe[46]) / (dataTempe[45] - dataTempe[47]);
        }
    else if (fiberLoc <= dataTempe[49]){
        FiberTemperature = dataTempe[46] - (dataTempe[47] - fiberLoc) *
        (dataTempe[46] - dataTempe[48]) / (dataTempe[47] - dataTempe[49]);
        }
    else   {
        opserr << "FireFiberSection::determineTemp -- fiber loc is out of the
        section at the bottom end \n";
        }
    static Vector returnedTemperature(2);
    returnedTemperature(0) = FiberTemperature;
    returnedTemperature(1) = FiberTempMax;
    return returnedTemperature;
}
```

C.1.2 Function *calculateC*

The following pages show the C++ code for calculating the shift in NA of a section.

```
double

FireFiberSection::calculateC(Vector &dataMixed)
{
	DataMixed = dataMixed;
	double stiffnessData[2];    //declare storage array for stiffness
	stiffnessData[0] = 0.0;
	stiffnessData[1] = 0.0;
	Vector dataTV;
	double fiberLocs[10000];
	double fiberArea[10000];

	if (sectionIntegr != 0){
		sectionIntegr->getFiberLocations(numFibers, fiberLocs);
		sectionIntegr->getFiberWeights(numFibers, fiberArea);
	}
	else
	{
		for (int i = 0; i < numFibers; i++) {
			fiberLocs[i] = matData[2 * i];
			fiberArea[i] = matData[2 * i + 1];
		}
	}

	for (int i = 0; i < numFibers; i++) {
		// initializing material strain and set it
		UniaxialMaterial *theMat = theMaterials[i];
		double secant = 0;
		double tangent = 0.0;
		double ThermalElongation = 0.0;
		double FiberTemperature = 0;
		double FiberTempMax = 0;

		if (fabs(DataMixed(1)) <= 1e-10 && fabs(DataMixed(49)) <= 1e-10)
		{
			FiberTemperature = 0;
			FiberTempMax = 0;
		}
		else
		{
			//calculate the fiber temp, T=T1-(Y-Y1)*(T1-T2)/(Y1-Y2)
			Vector TempV = this->determineFiberTemperature(DataMixed,
			fiberLocs[i]);
			FiberTemperature = TempV(0);
			FiberTempMax = TempV(1);
		}
		// get the data from thermal material
		static Vector tData(4);
		static Information iData(tData);
		tData(0) = FiberTemperature;
		tData(1) = tangent;
		tData(2) = ThermalElongation;
		tData(3) = FiberTempMax;
		iData.setVector(tData);
		theMat->getVariable("ElongTangent", iData);
		tData = iData.getData();
		ThermalElongation = tData(2);
		tangent = tData(1);
		Fiber_Tangent[i] = tangent;
		double y = fiberLocs[i];
		double A = fiberArea[i];
		double strain = theMat->getStrain();     //gives mechanical strain
		double stress = theMat->getStress();
```

```
            if (strain == 0) {

                  secant = theMat->getInitialTangent();
            }
            else {
                  secant = abs(stress / strain);
            }
            double EA = secante * A;
            double EAy = EA * y;
            stiffnessData[0] += EA;
            stiffnessData[1] += EAy;
      }
      double c = stiffnessData[1] / stiffnessData[0]; //EAy/EA
      return c;
}
```

C.2 Code of functions in the class $FireEl$

C.2.1 Function $getTangentStiff$

The C++ code for calculating and updating the stiffness matrix in the function $getTangentStiff$ is presented in the following page.

```
const Matrix&
FireEl::getTangentStiff()
{
	this->updateC();	//use the new NA position
	double L = crdTransf->getDeformedLength();//get deformed length
	//ELASTIC STIFFNESS MATRIX
	//assemble the basic (3x3) compressed form of the elastic stiffness matrix kb
	static Matrix kb(3, 3);
	kb.Zero();//initialise
	//get section properties from FireFiberSection
	double EA = ((FireFiberSection*)theSections[0])->getEA();
	double EI = ((FireFiberSection*)theSections[0])->getEI();
	//calculate entries;
	double EAoverL = EA/L;			// EA/L
	double EIoverL2 = 2.0*EI/L;		// 2EI/L
	double EIoverL4 = 4.0*EI/L;		// 4EI/L
	//populate kb in its basic form:
	kb(0, 0) = EAoverL;
	kb(1, 1) = kb(2, 2) = EIoverL4;
	kb(2, 1) = kb(1, 2) = EIoverL2;
	//create operator Qk which shifts stiffness matrix in basic form
	Matrix Qb(3, 3);
	Qb.Zero();	//set all to 0
	Qb(0, 0) = Qb(1, 1) = Qb(2, 2)  =1;
	Qb(0, 1) = c;
	Qb(0, 2) = -c;
	//initialise k_12 : stiffness around reference axis, and include shift
	Matrix kb_12(3, 3);
	kb_12.Zero();
	kb_12.addMatrixTripleProduct(1, Qb, kb, 1);	//transform basic matrix
	//GEOMETRIC STIFFNESS MATRIX
	//is not in a compressed form, but already in a local form : 6x6
	//get load vector from FireFiberSection
	//for node A
	const Vector resultant = ((FireFiberSection*)theSections[0])-
	>getStressResultant();
	double AxialForce1 = resultant(0);
	double Moment1 = resultant(1);
	//for node B
	const Vector resultant2 = ((FireFiberSection*)theSections[numSections - 1])-
	>getStressResultant();
	double Moment2 = resultant2(1);
	double AxialForce2 = resultant2(0);
	//use average axial force
	double AxialForceAvg = (AxialForce1 + AxialForce2) / 2;
	q(0) = AxialForceAvg;
	q(1) = -Moment1;
	q(2) = Moment2;
	//now shift the loads around new neutral axis
	q(1) -= AxialForceAvg*c;
	q(2) += AxialForceAvg*c;
	// Add effects of element loads, q = q(v) + q0 and include the shift
	q(0) += q0[0];
	q(1) += q0[1]-q0[0] * c;
	q(2) += q0[2] +q0[0] * c;
	//COMBINE ELASTIC AND GEOMETRIC STIFFNESS MATRIX
	K =  (crdTransf)->getGlobalStiffMatrix(kb_12, q, c);
	//K =  (crdTransf)->getGlobalStiffMatrix(kb, q, c);
	// this function calculates geometric stiffness matrix,
	//then adds it to the elastic stiffness matrix
	//and then transforms it to global coord.
	return K;

	}
```

C.2.2 Function *addLoad*

The function *addLoad* is shown in the following pages. This function can handle different load types and shifts them accordingly.

```
int
FireEl::addLoad(ElementalLoad *theLoad, double loadFactor)
{
	this->updateC();
	int type;	//type of load: 3 options
	const Vector &data = theLoad->getData(type, loadFactor);	//get the load and
	the loadfactor for that iteration
	double L = crdTransf->getDeformedLength();
	//mechanical beam loads: [begin]
	if (type == LOAD_TAG_Beam2dUniformLoad) {
		double wt = data(0)*loadFactor;  // Transverse (+ve upward)
		double wa = data(1)*loadFactor;  // Axial (+ve from node I to J)
		double V = 0.5*wt*L;
		double M = V * L / 6.0; // wt*L*L/12
		double P = wa * L;
		// Reactions in basic system
		p0[0] -= P;
		p0[1] -= V;
		p0[2] -= V;
		// Fixed end forces in basic system - named fb in thesis of jfv
		q0[0] -= 0.5*P;
		q0[1] -= M;
		q0[2] += M;
	}
	else if (type == LOAD_TAG_Beam2dPointLoad) {
		double P = data(0)*loadFactor;
		double N = data(1)*loadFactor;
		double aOverL = data(2);
		if (aOverL < 0.0 || aOverL > 1.0)
			return 0;
		double a = aOverL * L;
		double b = L - a;
		// Reactions in basic system
		p0[0] -= N;
		double V1 = P * (1.0 - aOverL);
		double V2 = P * aOverL;
		p0[1] -= V1;
		p0[2] -= V2;
		double L2 = 1.0 / (L*L);
		double a2 = a * a;
		double b2 = b * b;
		// Fixed end forces in basic system
		q0[0] -= N * aOverL;
		double M1 = -a * b2 * P * L2;
		double M2 = a2 * b * P * L2;
		q0[1] += M1;
		q0[2] += M2;
		}
	//mechanical beam loads [end]
	//thermal loads [begin]
	else if (type == LOAD_TAG_Beam2dThermalAction) {
		counterTemperature = 1;
		//intialising the temperature load vector
		q0Temperature[0] = 0.0;
		q0Temperature[1] = 0.0;
		q0Temperature[2] = 0.0;
		//organise temperature data into a 50 point array
		//jfv25 change to 25 points[begin]
		Vector dataMixV(50);
		int t = getTag();
		for (int m = 0; m < 25; m++) {
			dataMixV(2 * m) = data(2 * m); //Linear temperature interpolation
			dataMixV(2 * m + 1) = data(2 * m + 1);
```

```
                }
                for (int m = 0; m < 50; m++) {
                        dataMix[m] = dataMixV(m);   //transferrring the datatvector into
                        the double array that stores the temp //jfv

                }

                const Vector &s1 = ((FireFiberSection*)theSections[0])-
                >getTemperatureStress(dataMixV);    //contributed by ThermalElongation
                double ThermalForce1 = s1(0);
                double ThermalMoment1 = s1(1);
                const Vector &s2 = ((FireFiberSection*)theSections[numSections-1])-
                >getTemperatureStress(dataMixV);    //contribuited by ThermalElongation
                double ThermalForce2 = s2(0);
                double ThermalMoment2 = s2(1);

                q0Temperature[0] += (ThermalForce1+ ThermalForce2)/2;
                q0Temperature[1] -= ThermalMoment1;
                q0Temperature[2] += ThermalMoment2;

                q0Temperature[1] += q0Temperature[0] * c;//loadshift
                q0Temperature[2] -= q0Temperature[0] * c;
        }

else {
                opserr << "FireEl::addLoad(double) -- load type " << theLoad-
                >getClassType()
                        << "unknown for element with tag: " << this->getTag() << "\n";

                return -1;
        }

        // set up the shift operator in the basic system, called Qb as in
        q0[1] += q0[0] * c;  //loadshift
        q0[2] -= q0[0] * c;

        return 0;
}
```

List of References

Ansys (2016). ANSYS.

Bailey, C. (1998). Development of computer software to simulate the structural behaviour of steel-framed buildings in fire. *Computers & Structures*, vol. 67, no. 6, pp. 421–438. ISSN 0045-7949.
Available at: https://www.sciencedirect.com/science/article/pii/S0045794998000960

Bailey, C.G. (2004). Membrane action of slab/beam composite floor systems in fire. *Engineering Structures*, vol. 26, no. 12, pp. 1691–1703. ISSN 01410296.

Benedetti, A. and Mangoni, E. (2007). Analytical prediction of composite beams response in fire situations. *Journal of Constructional Steel Research*, vol. 63, no. 2, pp. 221–228.

Bhatti, M.A. (2016). *Advanced topics in finite element analysis of structures*. John Wiley & Sons. ISBN 978-0-471-64807-9.

Bisby, L., Gales, J. and Maluk, C. (2013). A contemporary review of large-scale non-standard structural fire testing. *Fire Science Reviews*, vol. 2, no. 1, pp. 1–27.

Bresler, B., Iding, R. and Nizamuddin, Z. (1977). *FIRES RC II - A computer program for the fire response of structures - reinforced concrete frames (revised version)*. Report NO. UCB FRG 77-8, University of California, Berkeley.

BSI (2004a). *EN 1992-1-1: Eurocode 2: Design of concrete structures - Part 1-1: General rules and rules for buildings*, vol. 1. Britisch Standards Institute, London.

BSI (2004b). *EN 1992-1-2 : Eurocode 2: Design of concrete structures - Part 1-2: General rules - Structural fire design*, vol. 2. Britisch Standards Institute, London.

BSI (2005). *EN 1993-1-2: Eurocode 3: Design of steel structures - Part 1-2: General rules - Structural fire design*, vol. 2. Britisch Standards Institute, London. ISBN 5947185067.

Buchanan, A.H. and Abu, A.K. (2017). *Structural design for fire safety*. 2nd edn. Wiley, Chichester.

Burgess, I. (2015). Vulcan.

Burgess, I.W., Huang, Z. and Plank, R.J. (2000). Three-dimensional analysis of composite steel-famed buildings in fire. *Journal of Structural Engineering*, vol. 126, no. 3, pp. 389–397.

Cedeno, G., Varma, A. and Gore, J. (2011). Predicting the standard fire behavior of composite steel beams. In: *Composite Construction in Steel and Concrete Conference VI*, pp. 642–656. ISBN 9780784411421.
Available at: http://ascelibrary.org/doi/abs/10.1061/41142(396)53

Clifton, G. and Abu, A. (2014). *Modifications to the application of the SPM:2006 edition and application to C/VM2*. Report, Auckland University, Auckland.

Clifton, G.C. (2006). *Design of composite steel floor systems for severe fires*. Report R4-131, HERA, Manukau City.

Cook, R.D., Malkus, D.S., Plesha, M.E. and Witt, R.J. (2001). *Concepts and application of finite element analysis*. 4th edn. WI: John Wiley & Son, Madison.

COST (2014). *Benchmark studies experimental validation of numerical models in fire engineering*. CTU Publishing House, Cech Technical University, Prague. ISBN 9788001054437.

Dassault Systeme Simulia (2016). ABAQUS.

Duthinh, D. (2014). NIST Technical Note 1842 Structural Design for Fire : A Survey of Building Codes and Standards. Tech. Rep..

Franssen, J.M. (2005). SAFIR : A thermal/structural program modelling structures under fire. *Engineering Journal, AISC*, vol. 42, no. 3, pp. 143–158.

Franssen, J.M., Cooke, G.M.E. and Latham, D.J. (1995). Numerical simulation of a full scale fire test on a loaded steel framework. *Journal of Constructional Steel Research*, vol. 35, pp. 377–408.

Gillie, M. (2009). Analysis of heated structures: Nature and modelling benchmarks. *Fire Safety Journal*, vol. 44, no. 5, pp. 673–680. ISSN 03797112.

Gorse, C. and Sturges, J. (2017). Not what anyone wanted: Observations on regulations, standards, quality and experience in the wake of Grenfell. *Construction Research and Innovation*, vol. 8, no. 3, pp. 72–75. ISSN 2045-0249.
Available at: `https://doi.org/10.1080/20450249.2017.1368260`

ISO (1999). *ISO 834 Fire-resistance tests - Elements of building construction Parts 1-12*. International Organization for Standardization, Geneva.

Iu, C.K., Chan, S.L. and Zha, X.X. (2005). Nonlinear pre-fire and post-fire analysis of steel frames. *Engineering Structures*, vol. 27, pp. 1689–1702.

Jeffers, A.E. and Sotelino, E.D. (2012). An efficient fiber element approach for the thermo-structural simulation of non-uniformly heated frames. *Fire Safety Journal*, vol. 51, pp. 18–26. ISSN 0379-7112.
Available at: `http://dx.doi.org/10.1016/j.firesaf.2012.02.002`

Jiang, J., Jiang, L., Kotsovinos, P., Zhang, J. and Usmani, A. (2015). OpenSees software architecture for the analysis of structures in fire. *Journal of Computing in Civil Engineering*, vol. 29, no. 1, pp. 1–13.

Jiang, J. and Usmani, A. (2013). Modeling of steel frame structures in fire using OpenSees. *Computers & Structures*, vol. 118, pp. 90–99. ISSN 0045-7949.
Available at: `https://www.sciencedirect.com/science/article/pii/S0045794912001800`

Jiang, J., Usmani, A. and Li, G.-Q. (2014). Modelling of steel-concrete composite structures in fire using OpenSees. *Advances in Structural Engineering*, vol. 17, no. 2, pp. 249–264. ISSN 13694332.

Jiang, L. (2017). OpenSees for fire, [Online].
Available at: `http://openseesforfire.github.io/`

Jiang, L. and Usmani, A. (2018). Computational performance of beam-column elements in modelling structural members subjected to localised fire. *Engineering Structures*, vol. 156, pp. 490–502. ISSN 0141-0296.
Available at: `https://www.sciencedirect.com/science/article/pii/S0141029617320552`

Kloos, M. (2017). *An investigation into the structural behaviour of a novel cellular beam structure in fire*. Master's thesis, Stellenbosch University.

Lamont, S. (2001). *The behaviour of multi-storey composite steel framed structures in response to fires*. Ph.D. thesis, University of Edinburgh.

Law, M. (1981). Designing fire safety for steel - recent work. In: *ASCE Spring Convention*.

Lennon, T. (2011). *Structural fire engineering*. ICE Publishing.

LSTC (2014). LS-DYNA.

Marx, H. (2018). *Thermal behaviour of a novel cellular beam structural system in fire*. Master's thesis, Stellenbosch University.

McGuire, W., Gallagher, R.H. and Ziemian, R.D. (2000). *Matrix structural analysis*. 2nd edn. John Wiley & Sons, New York.

McKenna, F. (1997). *Object-oriented finite element programming: frameworks for analysis, algorithms and parallel computing*. Ph.D. thesis, Univeristy of California, Berkeley.

NIST (2005). Final report on the collapse of the World Trade Center Towers. Tech. Rep., NCSTAR1, Washington DC.

Prokon SCL (2015). Prokon Frame. Tech. Rep., Prokon Software Consultants Ltd., Pretoria.

Quintiere, J.G. (2017). *Principles of fire behavior*. 2nd edn. Taylor& Francis Group, CRC Press, Boca Raton. ISBN 9781498735629.

Rackauskaite, E., Kotsovinos, P. and Rein, G. (2017). Model parameter sensitivity and benchmarking of the explicit dynamic solver of LS-DYNA for structural analysis in case of fire. *Fire Safety Journal*, vol. 90, pp. 123–138. ISSN 03797112.
Available at: `http://dx.doi.org/10.1016/j.firesaf.2017.03.002`

Sanad, A., Rotter, J., Usmani, A. and O'Connor, M. (2000). Composite beams in large buildings under fire - numerical modelling and structural behaviour. *Fire Safety Journal*, vol. 35, no. 3, pp. 165–188. ISSN 03797112.

Stern-Gottfried, J. and Rein, G. (2012). Travelling fires for structural design - Part I: Literature review. *Fire Safety Journal*, vol. 54, pp. 74–85. ISSN 0379-7112.
Available at: `https://www.sciencedirect.com/science/article/pii/S0379711212000756`

Thomas, P., Bullen, M., Quintiere, J. and McCaffrey, B. (1980). Flashover and instabilities in fire behavior. *Combustion and Flame*, vol. 38, pp. 159–171. ISSN 0010-2180.
Available at: `https://www.sciencedirect.com/science/article/pii/0010218080900486`

TNO DIANA (2016). DIANA.

Usmani, A., Drysdale, D., Rotter, J., Sanad, A., Gillie, M., Lamont, S., O'Conner, M., O'Callaghan, D., Elghazouli, A., Richardson, A., Izzuddin, B., Bailey, C., Newman, G. and Building (2000). PIT Project: Behaviour of steel framed structures under fire conditions. Tech. Rep., University of Edinburgh, Edinburgh.

Usmani, A., Rotter, J., Lamont, S., Sanad, A. and Gillie, M. (2001). Fundamental principles of structural behaviour under thermal effects. *Fire Safety Journal*, vol. 36, no. 8, pp. 721–744. ISSN 03797112.

Wainman, D. and Kirby, B. (1989). *Compendium of UK standard fire test data (unprotected structural steel - 2)*. 1st edn. Britisch Steel Corporation, Swinden Laboratories, Rotherham. ISBN RS/RSC/S10328/1/87/B.

Walls, R., Viljoen, C., de Clerq, H. and Clifton, G. (2017). Reliability analysis of the slab panel method (SPM) for the design of composite steel floors in severe fires. *Journal of Structural Fire Engineering*, vol. 8, no. 1, pp. 84–103.

Walls, R.S. (2016). *A beam finite element for the analysis of structures in fire*. Ph.D. thesis, Stellenbosch University.

Walls, R.S., Viljoen, C. and Clercq, H.D. (2018). Analysis of structures in fire as simplified skeletal frames using a customised beam finite element. *Fire Technology*. ISSN 1572-8099. Available at: `https://doi.org/10.1007/s10694-018-0762-7`

Wang, Y., Burgess, I., Wald, F. and Gillie, M. (2012). *Performance-based fire engineering of structures*. 1st edn. Spon Press, New York.